AWS First Cloud AI Journey – **Project Plan**

TEEJ Team – FPT University – AGREEME
09/12/2025

Table of Contents

1 BACKGROUND AND MOTIVATION

1.1 EXECUTIVE SUMMARY

Customers are individuals or small user groups (freelancers, small business owners, administrative/legal staff, people who work with contracts every day) but without deep legal expertise. They need a convenient, fast, and accurate tool to check and analyze contracts in their daily work, but do not want to invest in a complex system, nor do they need multi-user management or large-scale storage. Their main goals are: to understand contracts quickly, reduce risk, receive clear editing suggestions, and save the cost of hiring lawyers.

Business objectives:

- Have an AI tool that instantly analyzes contracts, as easy to use as a personal legal assistant.
- Reduce the cost of hiring legal consultants whenever a new contract arises.
- Shorten the time needed to read/understand a contract from hours to just a few minutes.
- Increase confidence when signing, transacting, and negotiating.

Technical objectives & reasons for choosing AWS:

- Leverage Amazon Bedrock for deep text analysis, ensuring accuracy and scalability.
- Use a serverless architecture (Lambda, DynamoDB, S3) to optimize cost and performance for individual users.
- Use AWS Amplify to build/deploy the frontend quickly, with Hosting + CDN + WAF integrated directly in Amplify, without setting up a separate WAF or complex CI/CD pipeline.
- Monitor the entire system with CloudWatch to track logs, metrics, and errors in real time.
- Integrate AWS-standard security (Cognito, KMS) to protect sensitive contract data.
- Enable fast deployment and lightweight operations, suitable for a SaaS model for individual users.

Use cases:

1. **Instant contract content analysis**
    - Explain complex clauses in simple language.
    - Attach legal context and highlight unfavorable points.
2. **Risk detection & alerts**
    - List unbalanced clauses and potential legal risks.
    - Estimate risk level (low / medium / high).
3. **Clause editing suggestions & alternative wording**
    - Propose amendment or negotiation options aligned with the user's objectives.
4. **Contract summarization**
    - Automatically generate an executive summary for busy users.
5. **Generate new contracts from templates / on demand**
    - Create simple contracts such as lease, sale, and service agreements.
    - Support revising them according to real-life situations.

6. **“Fast – compact – convenient” on web/app**
    - Upload contract → receive analysis → download/compare the revised version.
    - No need for system administration, multi-tenant setup, or complex workflows.

Implementation services provided by the partner include:

- Designing a lightweight serverless architecture using Lambda, API Gateway, DynamoDB, and S3.
- Developing the frontend on AWS Amplify, using Hosting + CDN + WAF integrated directly in Amplify.
- Integrating Amazon Bedrock for contract analysis, risk detection, summarization, and edit suggestions.
- Applying AWS standard security: Cognito for authentication, KMS for data encryption.
- Setting up CloudWatch for system monitoring, logging, alerts, and performance tracking.
- Automating deployment via Amplify (built-in CI/CD), without requiring a complex pipeline.
- Training and handover so individual users can conveniently use the system as an AI legal assistant.

1.2 PROJECT SUCCESS CRITERIA

- Contract analysis accuracy ≥ 85%, based on internal testing and real user feedback.
- Contract analysis time ≤ 7 seconds after the user uploads the document.
- Frontend error rate < 1%, according to CloudWatch and Amplify Monitoring statistics.
- System uptime ≥ 99.9%, ensuring individual users can access the service at any time.
- Reduce users’ contract reading/understanding time by at least 60%, based on before/after surveys.
- User satisfaction score ≥ 4/5, based on NPS or in-app surveys after 7 days of use.
- No security incidents, including data leaks, unauthorized access, or misconfigured permissions (Cognito + WAF).
- Ability to handle ≥ 500 analysis requests per day without manual infrastructure scaling (serverless auto-scaling).

1.3 ASSUMPTIONS

Assumptions & technical prerequisites:

- Users agree to provide their contract documents (PDF/Docx) for the system to process via AI models.
- Amazon Bedrock maintains stable operations and delivers text-analysis APIs as expected.
- AWS Amplify Hosting continues to support integrated WAF.
- Users have a stable internet connection to upload documents and receive AI responses quickly.
- Core serverless services (Lambda, API Gateway, DynamoDB, S3) maintain AWS’s standard SLA.

External dependencies:

- Dependent on the accuracy of GenAI/LLM models from Bedrock (Claude, Llama, Titan, etc.) when analyzing legal content.
- Users comply with local legal regulations regarding the use of AI and processing of contract data.

Constraints:

- The project does not require a complex multi-tenant system, so the scope focuses on individual users/small groups.
- High security requirements because contracts are sensitive documents → data must be encrypted (KMS) and not retained longer than necessary.
- No standalone WAF deployment — only the WAF integrated in Amplify is used.
- Minimize long-term storage of original documents to ensure user privacy.

Potential risks:

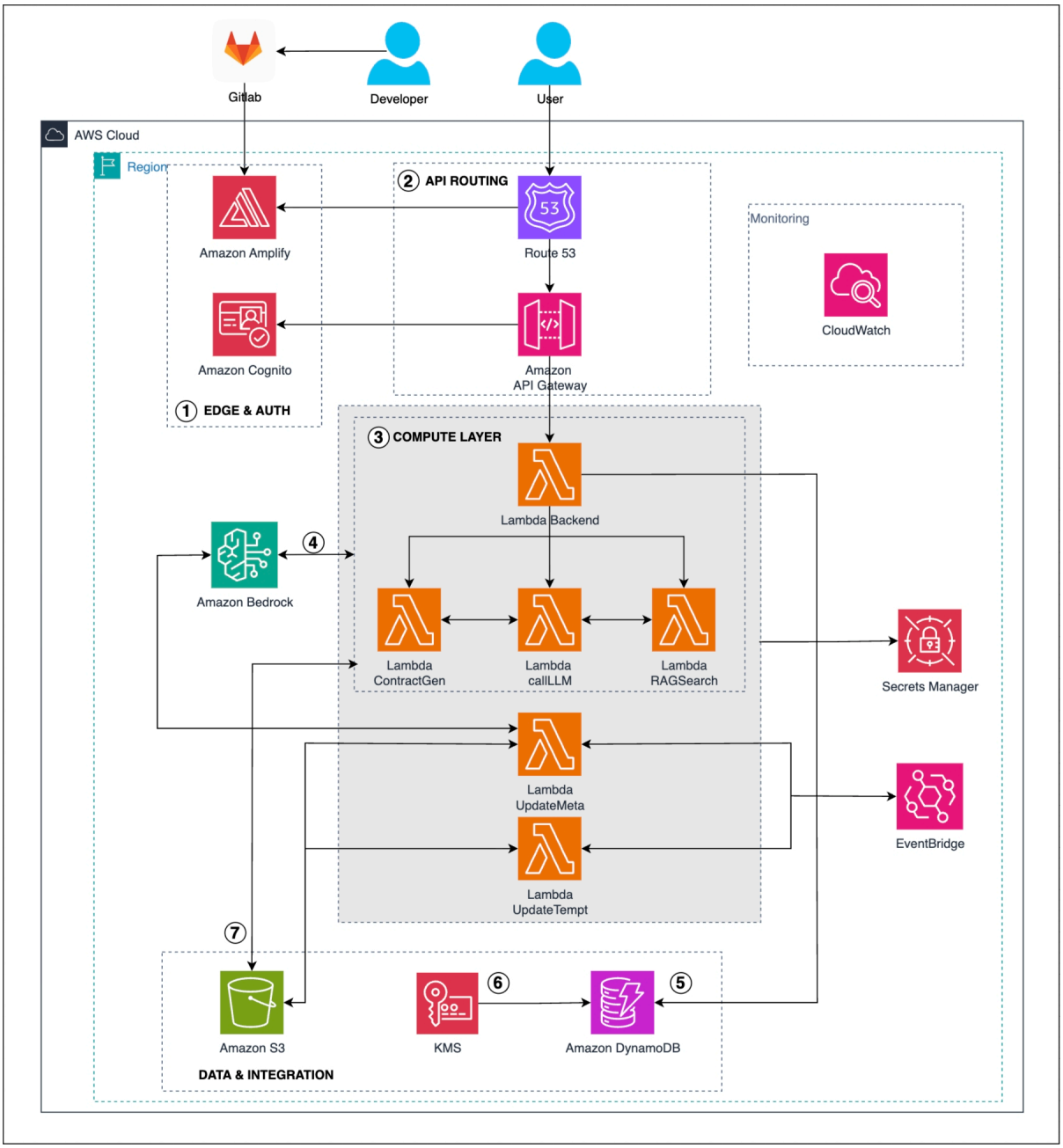
- AI accuracy risk: the model may misinterpret certain specific legal clauses → requires thorough testing and user warnings about AI limitations.
- Sensitive document risk: users may upload contracts containing highly sensitive personal information → requires clear data-handling policies.
- Cloud service dependency risk: if Bedrock or Amplify experiences an outage, the system will be temporarily unavailable.
- User growth & cost risk: although serverless scales automatically, AI models incur per-request costs → usage must be controlled and prompts optimized.

2 SOLUTION ARCHITECTURE / ARCHITECTURAL DIAGRAM

2.1 TECHNICAL ARCHITECTURE DIAGRAM

Technical Architecture Diagram

The proposed high-level technical architecture is designed as a fully serverless, secure, scalable, and operationally efficient solution built entirely on AWS managed services. The system is optimized for GenAI-driven document processing, contract generation, and RAG-based information retrieval. The architecture covers network flow, application components, data lifecycle, integration boundaries, security controls, and operational monitoring.



1. Network Infrastructure & Entry Points

- Amazon Route 53 serves as the public DNS entry point, routing traffic to the Amplify-hosted front-end application.
- AWS Amplify Hosting provides a globally distributed, secure, and auto-managed front-end hosting platform.
- Amplify automatically enables AWS WAF for the hosting distribution, providing baseline Layer-7 protection without requiring separate WAF deployments.
- All traffic to backend flows through Amazon API Gateway, enabling API routing, throttling, and authentication integration with Cognito.

2. Identity, Authentication & Access Control

- Amazon Cognito manages user pools and identity federation for user authentication.
- Cognito tokens (JWT) are validated by API Gateway before invoking Lambda functions.
- Fine-grained IAM roles restrict Lambda access to S3, Bedrock, DynamoDB, KMS, and EventBridge.

3. Backend Compute Layer

All backend logic is implemented using a modular, event-driven AWS Lambda microservices architecture.

Lambda Backend (Core API Handler): Orchestrates requests from API Gateway and routes them to downstream specialized Lambda functions.

Specialized Lambda Functions:

- Lambda ContractGen – Handles contract generation workflows using Bedrock foundation models.
- Lambda callLLM – Manages general LLM interactions (summaries, classification, transformations, etc.).
- Lambda RAGSearch – Executes embedding-based search, retrieval augmented generation, and knowledge lookup.
- Lambda UpdateMeta – Writes and updates document metadata in DynamoDB.
- Lambda UpdateTempt – Manages contract templates and associated updates.

All Lambda functions are fully serverless, auto-scaling, and instrumented with CloudWatch.

4. AI & LLM Integration

- Amazon Bedrock provides secure access to enterprise LLMs, including model inference for contract generation, embedding creation, and RAG flows.
- Bedrock API calls flow through tightly scoped IAM roles and do not require managing model infrastructure.

5. Storage, Data & Encryption

- Amazon S3 stores user-uploaded metadata, documents, generated contracts, and template files.
- Amazon DynamoDB holds user data.
- AWS KMS encrypts S3 objects, DynamoDB data, and secrets used across the system.

6. Eventing, Automation & Integration

- Amazon EventBridge coordinates asynchronous metadata updates, template changes, and system-level events.
- Enables decoupled, scalable system behavior without requiring a traditional messaging bus.

7. Monitoring & Operations

- Amazon CloudWatch captures logs, metrics, alarms, and distributed traces across Lambda, API Gateway, Amplify, and Bedrock interactions.
- Provides real-time observability for performance tuning, error detection, and operational reporting.
- Leverages Amplify's native CI/CD pipeline for automated deployment from GitLab.

8. Security Model

Security is enforced across all layers:

- Amplify-managed AWS WAF protects the front-end distribution by default.
- API Gateway throttling, authentication, and IAM–based backend access.
- DynamoDB policies enforce least privilege + encryption-at-rest via KMS.
- Serverless architecture removes the need to manage servers or networking ACLs.

Tools & AWS Services Proposed for the Project

Application Stack:

- AWS Amplify – Front-end hosting, GitLab CI/CD pipeline, WAF-protected distribution.
- React / Amplify Framework – Front-end UI.
- Amazon API Gateway – API management layer.
- AWS Lambda – Backend compute microservices.
- Amazon Bedrock – GenAI model inference and embeddings for RAG.
- Amazon DynamoDB – Metadata, templates, knowledge store.
- Amazon S3 – Document and template storage.
- AWS KMS – Encryption.
- Amazon EventBridge – Event routing and automation.

Monitoring & DevOps:

- Amazon CloudWatch – Logs, metrics, alarms, observability.
- GitLab – Source control & automated deployments via Amplify.
- Amplify CI/CD – Automatic build, test, deploy pipeline for front-end and backend updates.

Security:

- AWS WAF – Layer-7 protection for front-end distribution.
- AWS IAM – Role-based access for all resources.

2.2 TECHNICAL PLAN

Agreeme Team will develop deployment and configuration scripts using AWS Amplify configuration, environment settings, and Amplify-managed build and deployment workflows. This approach will allow for quick, consistent, and repeatable deployments across AWS environments while reducing manual configuration and operational overhead.

Some additional configuration, such as IAM role permissions, access policies to Amazon Bedrock, encryption key (KMS) usage, logging configurations, and data access scope definitions, may require customer review and approval and will follow the customer's established security and governance processes prior to implementation.

The solution will leverage AWS Amplify for frontend hosting, backend integration, authentication, and built-in WAF protection, with backend services implemented using AWS Lambda, Amazon DynamoDB, Amazon S3, and Amazon EventBridge.

Amazon CloudWatch will be used for centralized monitoring, logging, and operational visibility.

All critical application paths, including user authentication, contract upload, AI-driven contract analysis, and result presentation, will include extensive unit and integration test coverage.

2.3 PROJECT PLAN

Agreeme Team will adopt the Agile Scrum framework to deliver the project over 6 two-week sprints, spanning a total duration of 12 weeks. This delivery approach enables continuous delivery, early feedback, and iterative improvements, while ensuring alignment with the project's business and technical objectives.

Sprint Structure

Each sprint will include the following standard Scrum ceremonies:

- Sprint Planning
- Development and Testing
- Sprint Review
- Sprint Retrospective

Incremental and usable features will be delivered at the end of every sprint, ensuring visible progress, early validation, and the ability to adapt based on stakeholder feedback throughout the engagement.

Stakeholder Participation

Key stakeholders from the customer team are expected to:

- Attend Sprint Reviews to review delivered functionality and provide feedback.
- Participate in Sprint Retrospectives to suggest improvements and refinements.
- Review and approve major architectural, security, or IAM-related changes, as required.

Proposed Team Responsibilities

1. **Agreeme Team Responsibilities**

   Agreeme Team will be responsible for the end-to-end technical delivery of the solution, including:

   - Solution architecture and technical design.
   - Front-end and back-end development.
   - GenAI capability design and Amazon Bedrock integration.
   - Security controls and IAM configuration.
   - Deployment automation and environment management via AWS Amplify.
   - Testing, monitoring setup using Amazon CloudWatch, and technical documentation.
   - Knowledge transfer and project handover

2. **Customer Team Responsibilities**

   The customer team will support the project by providing:

   - Business requirement clarification and prioritization.

- Functional review and feedback during sprint reviews.
- Approval of security, IAM, and data governance decisions.
- Participation in User Acceptance Testing (UAT) and final sign-off

Communication Cadences

Communication will follow a structured cadence to ensure alignment and transparency:

- Daily stand-ups.
- Weekly progress sync with customer stakeholders.
- Bi-weekly sprint reviews and retrospectives.
- Ad-hoc architectural or security review sessions, as needed.

All project artifacts, sprint updates, and deliverables will be tracked and managed through a shared project management tool agreed upon by both parties.

Knowledge Transfer

Knowledge transfer sessions will be conducted by Agreeme Team during the final sprints and project closure phase, ensuring the customer team is fully enabled to operate and evolve the solution independently.

Knowledge transfer topics will include:

- System architecture and key design decisions.
- Deployment and release processes.
- Monitoring, logging, and troubleshooting using Amazon CloudWatch.
- Best practices for operating and extending the solution.
- GenAI prompt management and optimization guidelines.

Deliverables will include:

- Architecture diagrams
- Deployment and operational runbooks
- Operational guides
- Source code documentation

2.4 SECURITY CONSIDERATIONS

The security architecture of the solution is designed in accordance with AWS Shared Responsibility Model and AWS security best practices. Security controls are applied consistently across access management, infrastructure, data protection, detection, and incident management.

1. Access Security

Agreeme Team will enforce least-privilege access using AWS IAM roles and policies for all services and Lambda functions.

Amazon Cognito will be used for end-user authentication, with:

- Secure password policies
- Token-based authentication (JWT)
- Optional MFA configuration where applicable

Administrative access to AWS accounts will be restricted to authorized personnel only.

MFA will be enabled for all privileged AWS account access, including root and administrator accounts.

Secrets and credentials will not be hard-coded and will be managed through AWS-managed mechanisms and encrypted using KMS.

2. Infrastructure Security

The architecture is fully serverless, eliminating the need for operating systems, patch management, or network-level exposure.

AWS Amplify Hosting will provide front-end distribution with built-in AWS WAF protection, mitigating common web-based attacks (e.g., XSS, SQL injection).

Amazon API Gateway will enforce:

- HTTPS-only endpoints
- Request throttling and rate limiting
- Authentication integration with Cognito

Network access to AWS services is restricted via service-level IAM boundaries rather than open network access.

Infrastructure will be provisioned using Infrastructure-as-Code to ensure consistent and auditable configurations.

3. Data Security

- All sensitive data in DynamoDB will be encrypted at rest using AWS KMS.
- Amazon S3 objects (documents, generated contracts) default encryption.
- Data access will be restricted based on IAM policies and application context.
- S3 bucket policies will enforce least privilege and prevent public access.
- Data retention and lifecycle policies will be applied to minimize long-term storage of sensitive contract data.

4. Detection & Monitoring

AWS CloudWatch will be configured for continuous monitoring of:

- Lambda execution errors and latency
- API Gateway access logs and metrics
- Application-level error rates

Logging will be enabled across all critical services to ensure traceability.

Alerts and alarms will be configured to notify the operations team of abnormal behavior or service degradation.

5. Incident Management & Response

Agreeme Team will define and document an incident response procedure, including:

- Incident classification and severity levels.
- Escalation paths and communication protocols.
- Containment, remediation, and recovery steps.

CloudWatch alarms will trigger early detection of incidents to reduce mean time to resolution (MTTR).

Logs and audit trails will be retained to support root cause analysis.

Incident playbooks and runbooks will be provided as part of the operational handover.

3 ACTIVITIES AND DELIVERABLES

3.1 ACTIVITIES AND DELIVERABLES

Project Phase	Timeline	Activities	Deliverables/Milestones	Total man-day
Assessment	Week 1–2	• Business & user requirement assessment • Define AI use cases (analysis, risk detection, suggestion) • High-level architecture & security design	• Requirement summary • Architecture diagram • Security baseline	10 man-day
Setup Base Infrastructure	Week 3–4	• AWS Amplify project setup • IAM roles & access model • S3 & DynamoDB provisioning • KMS encryption • CloudWatch logging & monitoring	• Base infrastructure ready • Monitoring & logging enabled	14 man-day
Setup Frontend & Authentication	Week 5-6	• Frontend hosting via Amplify • Cognito authentication • Enable Amplify-integrated WAF • Contract upload & basic UI	• Secure frontend deployed • Authentication & WAF enabled	14 man-day
Setup Backend Core & AI Integration	Week 7-8	• Lambda backend APIs • Amazon Bedrock integration • Contract parsing & basic AI analysis • Store results in DynamoDB	• AI-enabled backend services • Initial analysis flow	18 man-day
Advanced AI & Optimization	Week 9-10	• Risk detection & negotiation suggestions • RAG search over contract content • EventBridge async workflows • UX improvements	• End-to-end AI contract intelligence • Optimized user experience	18 man-day
Testing, Golive and Handover	Week 11-12	• Unit & integration testing • Security & performance validation • Production deployment • Documentation & knowledge transfer	• Production go-live • Handover & KT completed	12 man-day

3.2 OUT OF SCOPE

- Enterprise-level contract management workflows and integrations.
- Custom CI/CD pipelines, infrastructure-as-code tools, or standalone security services.
- Training or hosting custom AI models beyond Amazon Bedrock.
- Legal validation or replacement of professional legal services.
- Multi-region, mobile app development, or long-term managed operations.

3.3 PATH TO PRODUCTION

The initial delivery will focus on a targeted set of contract analysis use cases as defined in Section I, emphasizing AI-assisted contract review, risk identification, and suggestion generation. While the solution will be production-ready for limited usage, it will not yet cover all features required for large-scale or enterprise-wide adoption.

To transition fully into production, additional enhancements will be required, including:

- Further hardening of security and access controls, including fine-tuning IAM policies and data access scopes.
- Expanded error handling, monitoring, and alerting to cover failure scenarios across AI inference, asynchronous processing, and data pipelines.
- Extended testing coverage, including performance, load, and resilience testing for peak usage scenarios.
- Optimization of AI prompts, response quality, and cost controls to support sustained daily usage.

Production readiness will also require iterative operational tuning and user feedback incorporation to achieve optimal reliability, scalability, and operational excellence.

4 EXPECTED AWS COST BREAKDOWN BY SERVICES

Services	Cost/Month (USD)
Amazon S3	1.80
Amazon API Gateway	0.05
Amazon DynamoDB	4.02
AWS Secrets Manager	1.08
Amazon Route 53	2.04
Amazon Cognito	1.00
AWS Amplify	16.25
Amazon CloudWatch	0.53

AWS Lambda	0.01
Amazon Bedrock	1.13
Total	**27.91**

Architectural Layer	**Total Cost (USD)**	**Cost Percentage**
Compute & Api	$0.06	0.2%
Data & Storage	$5.82	20.9%
Ai & Security	$3.21	11.5%
Async & Monitoring	$18.82	67.5%
Total	**$27.91**	**100%**

4.1 KEY COST DRIVERS

The following AWS services are identified as the primary cost-generating factors, accounting for the majority of the estimated budget:

- **AWS Amplify ($16.25 / 58.2%)**: Frontend hosting, build minutes, CDN.
- **Amazon DynamoDB ($4.02 / 14.4%)**: Application data storage, metadata, RAG knowledge store.
- **Amazon Route 53 ($2.04 / 7.3%)**: DNS and Hosted Zone service.
- **Amazon S3 ($1.80 / 6.5%)**: Raw document and contract template storage.
- **Amazon Bedrock ($1.13 / 4.1%)**: AI model costs for analysis and embedding generation.

4.2 COST OPTIMIZATION

To ensure cost efficiency during operation, we will implement the following optimization strategies:

- Optimize Amplify: Limit build minutes, streamline data transfer, and remove unused environments.
- Optimize DynamoDB: Apply right-sizing (auto-scaling/on-demand) and efficiently manage cold data storage.
- Optimize Bedrock: Implement token usage caps, shorten prompts, and utilize batch document processing.
- Control Route 53 & CloudWatch: Prevent unnecessary DNS records and tighten log retention policies.
- Maximize Free-tier: Fully leverage free AWS services (e.g., Lambda, low API Gateway usage).

5 TEAM

Partner Project Team

Name	Role	Email / Contact Info
Nguyễn Trí Dũng	AI Developer	nt.dung1297@gmail.com
Lê Minh Tuấn	Frontend	tuanlmse184475@fpt.edu.vn
Nguyễn Minh Nhật	Backend, AI Developer	nhatm2400@gmail.com
Trần Thị Minh Anh	Frontend	tranthiminhanh25@gmail.com

6 RESOURCES & COST ESTIMATES

Resource	Responsibility	Rate (USD) / Hour
Solution Architects	- Define overall system architecture (frontend, backend, AI, databases, security, integrations). - Choose cloud services and design scalable, secure, cost-effective infrastructure. - Align technical design with business goals (AI contract review, legal risk analysis). - Create high-level diagrams, standards, and non-functional requirements (availability, latency, compliance). - Coordinate between stakeholders, software engineers, AI engineers, and legal experts.	2.3
Software Engineers	- Implement web applications (UI/UX, APIs, authentication, user management, billing, etc.). - Integrate AI services (Bedrock/LLM APIs) into the backend and frontend flows. - Handle data storage, contract upload/download, history, and versioning. - Ensure performance, reliability, tests, and deployment pipelines follow the architecture.	0.7
AI Engineers	- Design and fine-tune AI models for contract understanding, clause classification, and risk scoring. - Build prompt pipelines / RAG over legal corpus and evaluate AI quality with legal benchmarks. - Implement logic for highlighting good/bad clauses and generating edit suggestions. - Optimize latency, cost, and safety of AI calls; monitor model performance in production.	0.7

Phase	Solution Architect	Software Engineer	AI Engineer	Total Hours
Assessment	Lead business & user requirement workshops; define AI use cases; design high-level architecture & security baseline.	Provide input on web stack, APIs, deployment approach; estimate implementation effort.	Define AI capabilities (analysis, risk, suggestion), data sources, and evaluation metrics.	**80 h**
Setup Base Infrastructure	Design cloud architecture (VPC, serverless pattern, security); review IAM and encryption strategy.	Set up Amplify project; provision S3, DynamoDB, IAM roles; enable CloudWatch logging/monitoring.	Specify storage needs for legal corpus, embeddings, and AI outputs; validate infra for future ML workloads.	**112 h**
Setup Frontend & Authentication	Validate UX flows, auth model, and WAF rules; ensure alignment with security & compliance.	Build UI in Amplify, integrate Cognito, implement contract upload & basic dashboard.	Define how AI results will be displayed (sections for good/bad clauses, suggestions); design response schema.	**112 h**
Backend Core & AI Integration	Design Lambda/API contracts, data model in DynamoDB, and Bedrock integration pattern.	Implement Lambda APIs, contract parsing, persistence, and connection to Bedrock endpoints.	Build prompts/RAG pipeline for legal analysis; implement initial clause classification & risk scoring; validate outputs with sample contracts.	**144 h**
Advanced AI & Optimization	Oversee end-to-end flow, define non-functional targets (latency, cost, reliability); approve optimization plan.	Implement advanced workflows (RAG search, EventBridge async jobs), UX improvements, and error handling.	Develop negotiation suggestions, refine models/prompts, tune cost & latency, set up monitoring for AI quality.	**144 h**

Phase	Solution Architect (h)	Software Engineer (h)	AI Engineer (h)	Total Hours	Total Cost (USD)
Assessment	40.0	20.0	20.0	80	**120.00**
Setup Base Infrastructure	33.6	56.0	22.4	112	**132.16**
Setup Frontend & Auth	22.4	67.2	22.4	112	**114.24**
Setup Backend Core & AI	28.8	57.6	57.6	144	**146.88**
Advanced AI Logic & Optimization	14.4	43.2	86.4	144	**123.84**

Hourly rates: Solution Architect = 2.3 USD/h, Engineer (Software/AI) = 0.7 USD/h

Total hours: 592 h (SA 139.2 h, SE 244.0 h, AI 208.8 h)

Total project cost (estimated): ≈ 637.12 USD