目次

職務経歴書

作成日: *2025年07月14日*

👨‍💻 黄 丞涓（ファン スンヨン）

フルスタックエンジニア | 6年経験 | Go・Kotlin・Java・TypeScript

📋 職務要約

6年間のフルスタック開発経験を持ち、Go・Kotlin・TypeScriptを核とした**サーバーサイドからフロントエンドまで一貫開発**が可能なエンジニア。特に**AI/LLM連携システム分野**でLangChain・プロンプトエンジニアリングを活用したインテリジェントなWebサービス開発と**マイクロサービスアーキテクチャ設計**に強みを持っています。**企画段階から運用保守まで一気通貫**で担当し、**ユーザーファースト**の視点から技術的意思決定をリード。**決済システム設計・実装**や**認証基盤構築**の豊富な経験を保有します。

🎯 **代表的な技術成果** - 🤖 **AI連携Webサービス開発** - LangChain + ChatGPT API統合による知的文書解析機能を実装 - 🧠 **プロンプトエンジニアリング** - Chain of Thought・役割指定・Few-shot学習技法適用 - 💳 **決済システム設計・実装** - Stripe連携によるサブスク・従量課金フロー構築 - 🚀 **マイクロサービス化** - モノリス分離で配布頻度**週1回→日複数回**に向上 - 📊 **システム性能改善** - DB最適化で処理時間**3時間→40分に短縮** - 🔄 **フルスタック開発体制** - 企画・設計・実装・運用まで一貫して担当

💼 スキルセット

🔧 バックエンド

- **Go (1.21)** - 2年/◎ / Echo, Gin, Wire
- **Kotlin** - 3年/◎ / Ktor, Exposed ORM
- **Java (17)** - 5年/◎ / Spring Boot, Micronaut
- **Python (3.11)** - 2年/◯

🌐 フロントエンド

- **TypeScript (5.2)** - 2年/◎ / 型安全性重視
- **Vue.js/Nuxt.js** - 2年/◯ / SPA開発
- **React.js/Next.js** - 1年/◯ / SSR/SSG

☁️ クラウド・インフラ

- **AWS** - 3年/◎ / Lambda, Aurora, S3, DynamoDB
- **Azure** - 2年/◯ / App Service, Functions
- **Docker** - 3年/◯ / コンテナ化, IaC

🗄️ データベース・ミドルウェア

- **PostgreSQL/MySQL** - 5年/◎ / クエリ最適化, インデックス設計
- **Redis** - 3年/◎ / キャッシュ戦略, 差分処理(SDIFF)
- **ElasticSearch** - 2年/◎ / 全文検索, 検索エンジン最適化
- **DynamoDB** - 1年/◯ / NoSQL設計, パフォーマンス調整

💡 保有スキル・知識

フルスタック・独立開発

- ✅ **フルスタック開発** - 企画・設計・実装・運用まで一貫して担当

- ✅ **プロダクト思考** - ユーザーファーストでのMVP設計・検証
- ✅ **決済システム** - Stripe連携、サブスク・従量課金実装
- ✅ **フロント+バック** - Vue.js/TypeScript + Go/Kotlin での同時開発

アーキテクチャ設計

- ✅ **クリーンアーキテクチャ** - DDD実践、レイヤード設計
- ✅ **マイクロサービス** - 独立したサービス設計、API Gateway
- ✅ **サーバーレス** - AWS Lambda, イベント駆動型設計
- ✅ **RESTful API** - OpenAPI仕様、バージョニング戦略

DevOps・CI/CD

- ✅ **IaC** - CloudFormation, SAM
- ✅ **CI/CD** - GitHub Actions
- ✅ **監視** - CloudWatch, Datadog, Sentry
- ✅ **テスト** - TDD, BDD, E2Eテスト

🏢 職務経歴

🤖 S社 ｜ フルスタックエンジニア

`2023年8月 - 現在`

🔍 企業・材料データベース検索システム＋AI連携開発

🎯 **主要成果:**

- 🤖 ChatGPT API統合による各種PDF文書解析・自社DB化機能を独力で設計・実装
- 📈 AI検索機能により検索精度向上とユーザー滞在時間150%増加

🎵 C社 ｜ フルスタックエンジニア

`2023年8月 - 2025年5月`

📱 AI音声プラットフォーム開発プロジェクト

🎯 主要成果:

- 💳 決済システムの設計・実装を主導（Stripe連携、サブスク・従量課金対応）
- 🔄 企画→開発→リリースまで一人で完結する開発体制を確立

🔐 I社 ｜ バックエンドエンジニア

`2022年3月 - 2023年7月`

📊 統合ログ管理システム - クラウド連携モジュール開発

🎯 主要成果:

- 📊 パフォーマンス改善により処理時間を82%短縮（3時間→40分）
- 📚 共通ライブラリ化により開発工数50%削減

🏢 U社 ｜ バックエンドエンジニア

`2021年2月 - 2022年2月`

🗺️ BtoB営業支援SaaSプロダクト開発

🎯 主要成果:

- 🔍 検索APIレスポンス時間80%改善（5秒→1秒以下）
- 📊 データベースストレージ使用率57%削減（70%→30%）

💻 H社 ｜ バックエンドエンジニア

`2020年2月 - 2021年2月`

🏪 オフィス家具メーカー向け受発注システム開発

🎯 **主要成果:**

- 🧪 テストカバレッジ112%向上（40%→85%）
- 🔄 手戻り率75%削減（20%→5%）

🎓 学歴・資格

項目	内容
学歴	大学卒業
語学	TOEIC 845点、アメリカ留学経験

🚀 今後のキャリアビジョン

1. **AI/検索システム専門性の深化**: 大規模検索エンジンとAI技術を融合した次世代プラットフォーム開発
2. **フルスタック技術リーダー**: Go・Kotlin・TypeScriptを軸としたクラウドネイティブアーキテクチャの設計・実装
3. **フルスタック開発体制の確立**: 企画から運用まで一気通貫で担当できる総合力を活かした価値創造
4. **チーム技術力向上への貢献**: 積極的なコミュニケーションと知識共有でチーム全体の開発効率化をリード
5. **継続的学習と技術革新**: 日々スキルを伸ばす努力を続け、最新技術動向をキャッチアップして実務に活用

📥 ダウンロードリンク

- **マークダウン**: resume-complete.md
- **HTML**: Releases タブで確認

📫 連絡先

メール: syhwang.web@gmail.com

最後までご覧いただき、ありがとうございました。

—

S社 詳細情報

🤖 S社 - AI連携企業・材料データベース検索システム開発

基本情報

項目	詳細
期間	2023年8月 - 現在
職種	フルスタックエンジニア
チーム規模	独立開発
役割	AIシステム設計 - 高度検索・AI文書解析・フルスタック開発

技術スタック

Backend

- **Kotlin** - メイン開発言語
- **Ktor** - Webフレームワーク
- **Exposed ORM** - DB操作
- **PostgreSQL** - メインDB
- **GraphQL** - API設計
- **Firebase Auth** - 認証基盤
- **Koin DI** - 依存性注入

Frontend

- **Next.js** - Reactフレームワーク
- **TypeScript** - 型安全開発
- **React Hook Form** - フォーム管理
- **Zod** - バリデーション
- **Recoil** - 状態管理
- **Apollo Client** - GraphQLクライアント
- **Shadcn/ui** - UIコンポーネント

Infrastructure & AI

- **AWS** - クラウドインフラ
- **Docker** - コンテナ化
- **OpenAI GPT-4** - AI文書解析
- **LangChain** - AI処理パイプライン
- **Document AI OCR** - OCR処理
- **PDF解析AI** - 文書処理

プロジェクト概要

> **AIファースト思考**で企業・材料データベースの検索体験を革新。**ChatGPT API統合**による各種PDF文書解析機能、**Document AI OCR**、**高度な検索システム**、**文書自動分類システム**の設計・実装を主導。PDF・画像文書の読み取りから自社データベース化まで一貫したインテリジェントなWebサービスを構築。

🔍 主な成果・実装内容

AI・機械学習システム実装

🤖 AI連携文書解析機能

機能	技術・実装内容
ChatGPT API統合	各種PDF文書の構造化データ抽出・自社DB化機能
LangChain活用	AI処理パイプライン構築・チェーン処理最適化
プロンプトエンジニアリング	Chain of Thought, 役割指定, Few-shot学習技法適用
Document AI OCR	PDF・画像文書の自動読み取り・テキスト化処理
文書管理システム	技術資料の読み取り・データベース統合管理
ヒューリスティック分類	ゼロショットプロンプティングによる文書自動分類システム
DX機能実装	文書処理自動化によるユーザー体験向上

🔍 高度検索システム設計

機能	技術・実装内容
検索アルゴリズム最適化	関連度スコアリング・ランキング調整
リアルタイム候補表示	入力補完・検索提案システム
複合条件検索	企業属性・材料特性の多次元フィルタリング

🏗️ フルスタックアーキテクチャ設計

バックエンド設計（Kotlin）

- **クリーンアーキテクチャ**: DDD実装、レイヤード設計による保守性向上
- **GraphQL API**: 型安全なスキーマ設計、DataLoader パターンでN+1解決
- **Exposed ORM**: PostgreSQL連携、トランザクション管理
- **Firebase認証**: JWT トークン検証、権限管理システム

フロントエンド開発（TypeScript）

- **Next.js 14**: SSR/SSG最適化、パフォーマンス向上
- **状態管理**: Recoil（グローバル）+ React Hook Form（フォーム）
- **UI/UX**: Shadcn/ui + Tailwind CSS、アクセシビリティ対応
- **型安全性**: 厳格なTypeScript、Zod バリデーション

🎯 独立開発・技術的意思決定

プロダクト企画・技術選定

- **要件定義**: ユーザーインタビューからAI機能仕様策定
- **技術選定**: パフォーマンス・保守性・拡張性を考慮した最適解選択
- **アーキテクチャ設計**: マイクロサービス分離、API設計

開発プロセス改善

- **CI/CD構築**: GitHub Actions、自動テスト・デプロイ
- **コード品質**: ESLint、Prettier、detekt による品質担保
- **監視・ログ**: 構造化ログ、エラートラッキング、パフォーマンス監視

📊 数値成果

+ 🤖 ChatGPT API統合による各種PDF文書解析・自社DB化機能の設計・実装を主導
+ 📄 ヒューリスティック・ゼロショット分類による文書処理DX機能を実現
+ 🚀 フルスタック開発体制を確立（企画から運用まで一貫して担当）
+ 📈 AI検索機能により検索精度向上とユーザー滞在時間150%増加

C社 詳細情報

🎵 C社 - AI音声プラットフォーム開発プロジェクト

基本情報

項目	詳細
期間	2023年8月 - 2025年5月
職種	フルスタックエンジニア
チーム規模	10名（エンジニア8名、デザイナー1名、PM1名）
役割	独立開発者 - 企画・バックエンド・決済システム・運用まで一人で担当

技術スタック

Backend

- **Go** - メイン開発言語
- **Echo** - Webフレームワーク
- **AWS Lambda** - サーバーレス
- **Aurora** - RDBクラスター
- **DynamoDB** - NoSQLデータベース

Frontend

- **Vue.js** - プログレッシブフレームワーク
- **Nuxt.js** - Vue.jsフレームワーク
- **TypeScript** - 型安全開発
- **Pinia/Vuex** - 状態管理
- **Tailwind CSS** - ユーティリティCSS

Infrastructure & Payment

- **AWS SAM** - Infrastructure as Code
- **Docker** - コンテナ化
- **Stripe** - 決済システム
- **Firebase** - 認証・リアルタイムDB
- **CloudWatch** - 監視・ログ

プロジェクト概要

> **ユーザーファースト開発**を実践し、プロダクト企画段階から技術選定・実装・運用まで一気通貫で担当。ビジネス要件をエンジニアリング視点で最適化し、ユーザーにとって最高の音声生成体験を提供するための技術的意思決定をリード。

🏗️ アーキテクチャ設計と実装

🏛️ クリーンアーキテクチャの厳格な実装

- **ドメイン駆動設計（DDD）**: ビジネスロジックの責任分離
- **レイヤード構造**: `domain` → `repository` → `service` → `handler`
- **依存性逆転の原則**: 疎結合による保守性向上

☁️ サーバーレスマイクロサービス

- **機能別Lambda関数**: 独立したサービス設計
- **イベント駆動型**: 非同期処理による高いスケーラビリティ
- **AWS SAM**: Infrastructure as Code による一貫した環境管理

🔧 フルスタック技術実装

🔍 バックエンド開発

技術領域	実装内容
依存性注入	Google Wireによるコンパイル時DI
コード生成	oapi-codegen, mockgen, sqlboiler
DB設計	Aurora（RDB）+ DynamoDB（NoSQL）のハイブリッド構成
認証基盤	Firebase Authentication + カスタムRBAC

🌐 フロントエンド開発

技術領域	実装内容
Vue.js/Nuxt.js	TypeScriptでの型安全なSPA開発
状態管理	Pinia 基盤 モダン状態管理 (Vuex 代替)

💳 決済システム設計・実装

💳 Stripe連携システム

機能	実装内容
決済方式	サブスクリプション・従量課金・ワンタイム決済
Webhook処理	決済状態の同期とエラーハンドリング
課金ロジック	使用量ベース課金システムの設計
ユーザー体験	決済フローのシームレスな統合

📊 数値成果

+ 💳 決済システムの設計・実装を主導（Stripe連携、サブスク・従量課金対応）
+ 🔄 企画→開発→リリースまで一人で完結する開発体制を確立
+ 💰 サーバーレス化によりインフラコストを30%削減
+ 🧪 テストカバレッジ90%以上を維持

I社 詳細情報

🔐 I社 - 統合ログ管理システム クラウド連携モジュール開発

基本情報

項目	詳細
期間	2022年3月 - 2023年7月
職種	バックエンドエンジニア
チーム規模	4名（サーバーサイドチーム）
役割	クラウドAPI連携、パフォーマンス最適化、アーキテクチャ改善

技術スタック

```
Java · Kotlin · Azure SDK · Microsoft Graph API · SalesForce API · Maven
```

🎯 課題と解決策

🚀 開発効率化

課題	解決策	結果
コード重複による開発効率低下	共通ライブラリ化（Maven Repository）	**開発工数50%削減**
バージョンアップ検証の工数増大	順次アップグレードフレームワーク開発	**検証工数を1/3に削減**

📊 パフォーマンス最適化

📊 大量ログ処理のパフォーマンス問題

課題: 大量ログ処理のパフォーマンス問題

解決策: - バッチ処理アルゴリズム最適化 - Visual VM + IntelliJ連携プロファイリング環境構築 - マルチスレッドプログラミングによる並行処理最適化

結果: **処理時間を3時間→40分に短縮（82%改善）**

🔧 技術的工夫

🔍 プロファイリング環境

- **Visual VM + IntelliJ**: 連携による詳細パフォーマンス分析
- **ボトルネック特定**: メモリ使用量・CPU使用率の可視化
- **最適化指針**: データ駆動な改善アプローチ

🔨 カスタムツール開発

- **Apex + Python**: 独自ログ収集ツール開発
- **自動化促進**: 手動作業の大幅削減
- **運用効率化**: 継続的な改善プロセス構築

📊 主な成果

+ 📊 パフォーマンス改善により処理時間を82%短縮（3時間→40分）
+ 📚 共通ライブラリ化により開発工数50%削減
+ 🔄 順次アップグレードフレームワークで検証工数を67%削減
+ 🔨 カスタムツール開発により運用自動化を実現

U社 詳細情報

🏢 U社 - BtoB営業支援SaaSプロダクト開発

基本情報

項目	詳細
期間	2021年2月 - 2022年2月
職種	バックエンドエンジニア
チーム規模	4名（アジャイル開発）
役割	API設計・開発、パフォーマンス改善、システム監視

技術スタック

```
Kotlin · Spring Boot · PostgreSQL · Redis · Elasticsearch · Azure · GraphQL
```

🚀 主な実装と成果

1️⃣ バッチ処理の最適化

📊 大容量バッチ処理の最適化

課題
大容量データのバッチ処理によるDBへの負荷集中

解決策 - **Redis SDIFF活用**: 差分比較処理による効率的なデータ処理 - **優先度別バッチ分割**: 負荷分散による処理性能向上 - **フォールバック実装**: メモリ障害時の安全な処理継続

結果
DBストレージ使用率 70% → 30% に削減（57%改善）

2️⃣ 検索APIのパフォーマンス改善

🔍 大規模データ検索の高速化

課題
大企業ユーザーの大量データで検索が遅延

解決策 - **Elasticsearch導入**: 全文検索による高速化 - **自動同期パイプライン**: RDB → Logstash → ES - **クエリ最適化**: must, should, term クエリの効率化

結果
レスポンスタイム 5秒 → 1秒以下（80%改善）

🔧 技術的取り組み

📊 データ処理最適化

- **Redis SDIFF**: 効率的な差分処理実装
- **バッチ処理**: 負荷分散とエラーハンドリング
- **メモリ管理**: 大容量データ処理の最適化

🔍 検索システム構築

- **Elasticsearch**: 高速全文検索システム
- **データ同期**: リアルタイム同期パイプライン
- **GraphQL API**: 型安全なAPI設計・実装

📊 主な成果

```
+ 🔍 検索APIレスポンス時間80%改善（5秒→1秒以下）
+ 📊 データベースストレージ使用率57%削減（70%→30%）
+ 🔴 Redis SDIFF活用による効率的な差分処理システム構築
+ 🔍 Elasticsearch導入による高速全文検索システム実現
```

H社 詳細情報

💻 H社 - オフィス家具メーカー向け受発注システム開発

基本情報

項目	詳細
期間	2020年2月 - 2021年2月
職種	バックエンドエンジニア
チーム規模	4名（スクラム開発）
役割	API実装、テスト設計、新人教育

技術スタック

```
Java · Groovy · Micronaut · Spock · H2 Database · PostgreSQL · Docker
```

📊 主な取り組みと成果

🧪 テスト環境構築・品質向上

🧪 包括的なテスト環境構築

取り組み	実装内容	成果
H2 Database活用	インメモリDBでの高速テスト実行	テスト実行時間大幅短縮
Spock BDDテスト	自然言語ライクなテスト記述	**テストカバレッジ 40% → 85%**
自動化テスト	CI/CDパイプラインとの統合	品質担保プロセス確立

🔄 チーム開発プロセス改善

🔄 継続的改善プロセス構築

技術共有文化の構築 - **Slackチャンネル提案・運用**: 技術情報の積極的な共有 - **知識の属人化防止**: チーム全体のスキル底上げ

品質管理プロセス導入 - **なぜなぜ分析**: 根本原因の特定と改善策立案 - **手戻り率大幅削減**: 20% → 5%（75%改善）

🔌 API開発・実装

🔍 Backend Development

- **Micronaut**: 軽量フレームワークでの高速API開発
- **RESTful API**: 設計原則に基づいた実装
- **PostgreSQL**: 効率的なデータベース設計

👨‍💻 Development Environment

- **Docker**: コンテナ化による一貫した開発環境
- **Groovy**: 簡潔なコード記述による生産性向上
- **スクラム開発**: アジャイルプロセスの実践

🌱 成長と学習

> **初級エンジニア**として、技術スキルと同時にチーム開発における**協調性**と**コミュニケーション能力**を向上。**テスト駆動開発（TDD）**の実践とテスト品質向上への貢献、**継続的改善のマインドセット**を構築。

📊 主な成果

+ 🧪 テストカバレッジ112%向上（40% → 85%）
+ 🔄 手戻り率75%削減（20% → 5%）
+ 👥 チーム内技術共有文化の構築・定着
+ ✅ 品質重視の開発プロセス確立