Akhil Reddy Pabbathi Reddy

Senior Java Full Stack Developer My Portfolio

PROFESSIONAL SUMMARY

- Developing and architecting enterprise-grade applications with 10+ years of experience in Java, J2EE, and Microservices architectures, delivering scalable, secure, and high-performance solutions across industries.
- Building and optimizing Full-Stack applications using Spring Boot, React.js, Angular, Node.js, and Vue.js, delivering responsive, user-friendly interfaces and maintaining seamless cross-platform compatibility.
- Architecting and implementing Microservices with Spring Boot, Spring Cloud, Kubernetes, and event-driven architectures, enabling modular scalability, fault tolerance, rapid deployment, and service autonomy.
- Leading and developing RESTful APIs, GraphQL APIs, SOAP Web Services, WebSockets, and asynchronous messaging using Apache Kafka, RabbitMQ, ActiveMQ, and JMS for reliable data transmission at scale.
- Designing and deploying highly scalable cloud-native solutions on AWS (EC2, Lambda, API Gateway, DynamoDB, S3, RDS, IAM, CloudWatch) and Azure (Functions, API Management, SQL, Blob, Cosmos DB), including multi-tenant SaaS systems with S3, CloudFront, and cost-optimized horizontal scaling.
- Automating cloud infrastructure with Terraform, AWS CloudFormation, Pulumi, and Ansible, applying infrastructure-as-code practices for consistent, version-controlled, and secure cloud deployments.
- Implementing robust CI/CD pipelines using Jenkins, GitHub Actions, GitLab CI, AWS CodePipeline, and tools like SonarQube, Black Duck, and Artifactory for fast, automated, and secure software releases.
- Optimizing and managing SQL & NoSQL databases including Oracle, PostgreSQL, MySQL, MongoDB, DynamoDB, Cassandra, Redis, and Elasticache by tuning queries, setting indexes, and ensuring high availability.
- Building and containerizing applications using Docker, Kubernetes (EKS, AKS, OpenShift), Helm, and Istio, supporting efficient deployments with auto-scaling, service mesh, health checks, and load balancing.
- Developing dynamic front-end applications using React.js, Redux, Angular, Vue.js, TypeScript, and Tailwind CSS, delivering responsive SPAs with pixel-perfect UIs and optimized rendering performance.
- Enhancing observability and proactive monitoring with Grafana, Kibana, ELK Stack, Prometheus, AWS CloudWatch, Fluentd, and alerting systems to improve system health, uptime, and incident recovery.
- Executing performance optimizations using JVM profiling, Heap Dump Analysis, GC tuning, thread analysis, and load testing with Apache JMeter for improved throughput, latency, and memory usage.
- Building event-driven architectures using CQRS, Saga Patterns, AWS SNS, SQS, Kinesis, Kafka Streams, and EventBridge for decoupled systems, real-time processing, and reliable messaging pipelines.
- Applying test-driven development (TDD) and automated testing using JUnit, Mockito, Cypress, Selenium, TestNG, Karma, Jasmine, and Protractor to ensure stability, fast feedback, and continuous integration.
- Leading Agile development teams with sprint planning, backlog grooming, retrospectives, and daily Scrum meetings, promoting collaboration, ownership, velocity improvement, and on-time deliveries.
- Engineering cloud-based AI/ML pipelines using AWS SageMaker, TensorFlow, PyTorch, OpenAI APIs, and Step Functions to enable predictive analytics, intelligent automation, and fraud detection in production.

Skills

Programming Languages:

Java 8/11/17, J2EE, Core Java, SQL, PL/SQL, Python, TypeScript, JavaScript (ES6+), Rust, Kotlin, Swift, Go, Shell Scripting, jQuery, AJAX

Frameworks & Libraries:

Spring (Boot, MVC, Cloud, Security, Batch, AOP, Data, Integration, REST, JDBC), Hibernate, Struts, Log4j, Quartz Scheduler, Apache Camel, React.js, Redux, React Native, Vue.js, AngularJS, Node.js, Express.js, GraphQL, Bootstrap, Tailwind CSS, DWR

Databases & Cloud Services:

PostgreSQL, Oracle (8i–12c), MySQL, MS SQL Server, MongoDB, DynamoDB, Cassandra, DB2, Redis, Elasticache, AWS (EC2, S3, RDS, Lambda, IAM, API Gateway, CloudFormation, KMS, SQS, SNS, CloudWatch, Elastic Beanstalk), Azure (Functions, API Management, Blob Storage, Cosmos DB, Azure SQL), Kubernetes, Docker, OpenShift, Terraform

Web & API Technologies:

JSP, Servlets, JDBC, RESTful APIs, SOAP Web Services, WSDL, JSON, XML, XSLT, WebSockets, JSTL, Dojo, Redux Toolkit, JSX-WS, GWT

Development Tools & CI/CD:

Eclipse, IntelliJ IDEA, NetBeans, Visual Studio Code, Apache Tomcat, WebLogic, WebSphere, JBoss, Nginx, Git, GitHub, Bitbucket, SVN, Jenkins, SonarQube, Terraform, Prometheus, Grafana, Kibana

EDUCATION

Andhra University

Bachelor's Degree in Computer Science and Engineering

Jul 2008 – Jun 2012 Visakhapatnam, IN

Mar 2023 – Present

Key Courses: Algorithms, Data Structures, Cloud Computing, Distributed Systems, Operating Systems, Computer Networks, Databases, Software Engineering, Object-Oriented Programming, Compiler Design, Artificial Intelligence, Web Technologies, Computer Architecture, Information Security, Machine Learning, Mobile Computing.

WORK EXPERIENCE

Java Full Stack Developer

Discover Financial | New York City, NY

At *Discover Financial*, I led the end-to-end development and optimization of a real-time transaction processing platform, engineered to handle millions of credit card and loan transactions daily with sub-millisecond response times. This platform significantly enhanced fraud detection, automated payment reconciliation, and strengthened risk scoring mechanisms, ensuring seamless and secure payment processing in compliance with PCI-DSS and SOX regulations.

- Designed and developed a high-performance financial transaction processing platform using Java 17, Spring Boot, Hibernate, and React with TypeScript, integrating OAuth2 and JWT security protocols, handling millions of transactions daily with sub-millisecond latency, and ensuring scalability, resilience, and fault tolerance.
- Integrated real-time payment systems with credit risk engines using Spring Boot, Kafka, Redis, and Al-driven fraud detection, improving risk scoring accuracy, compliance workflows, and reducing false positives. Utilized OpenTelemetry, monitoring, and test automation.
- Designed and deployed distributed microservices using Spring Cloud, Kubernetes, Docker, and Istio, implementing service discovery, circuit breakers, and zero-downtime deployments for secure, modular, and auto-scalable architecture.
- Developed low-latency fraud detection models using Apache Kafka Streams, AWS Lambda, SageMaker, and machine learning pipelines, increasing precision in anomaly detection, reducing fraudulent activities by 30%, and enabling continuous model retraining. Utilized log aggregation, feature flags, model accuracy metrics, CI logs.
- Engineered secure, RESTful and GraphQL-based API integrations with Visa, Mastercard, and PayPal using Spring Security, OAuth2, and API Gateway, reducing payment latency, boosting throughput, and maintaining full PCI-DSS, ISO 27001, and GDPR compliance. Enabled CORS, endpoint protection, secure cookie handling, and audit logs.
- Developed and optimized real-time transaction dashboards using Angular 14, RxJS, and Highcharts, improving UX responsiveness and lowering UI render latency by 35%. Included authentication states, session storage, route guards, and UX principles.
- Optimized PostgreSQL performance with partitioning, advanced indexing, and query parallelism, cutting reporting time by 50% and accelerating transactional batch workloads. Implemented deadlock prevention, table sharding, query queues, partition policies.
- Migrated legacy UI components from React to Angular 14 using Nx monorepos, improving modularity, scalability, and test coverage with Cypress and Jest. Ensured AOT build optimization, component reuse, and typed forms.
- Implemented asynchronous architecture using Kafka, SQS, and AWS Step Functions for transaction settlements, refunds, and disputes, enhancing throughput and scalability. Integrated message deduplication, SQS policies, and async error handling.
- Built and secured REST APIs with Spring Boot and WebSockets, enabling real-time updates, authentication using RBAC, and seamless integration with Angular apps. Ensured CSRF tokens, input sanitization, schema validation, and token refresh.
- Applied role-based access control (RBAC) and OAuth2 for internal systems, enhancing security posture and enforcing SOX and GDPR compliance policies. Established secure token handling, LDAP fallback, JWT expiry tracking.
- Built CI/CD pipeline with Jenkins, GitHub Actions, SonarQube, and Gradle, reducing deployment cycles by 60% and increasing release confidence and traceability. Managed artifact promotion, pipeline secrets, rollback notifications, pre-deployment.
- Developed backward-compatible adapters for legacy systems using REST and GraphQL, enabling data federation and modernization across core Discover services. Built schema translation logic, legacy data mapping, backward compatibility support.
- Deployed and orchestrated Docker containers on AWS EKS with HPA and ALB, improving uptime during peak transaction volumes and reducing resource cost by 25%. Added auto-healing strategies, logging sidecars, cross-zone load balancers.
- Automated RDS (PostgreSQL) cluster provisioning using AWS CDK with IAM, encryption, backups, and multi-AZ failover for 99.99% availability and disaster recovery. Monitored failover events, snapshot intervals, encryption keys, and replication lag.
- Implemented full-stack observability using Grafana, Kibana, CloudWatch, and ELK stack, enabling API latency tracking, alerting, and real-time service health visualization. Hooked into alert loops, dashboard refresh intervals, and KPI graphs.
- Enhanced query tuning and replicated read-heavy PostgreSQL workloads using CDK-managed Read Replicas, reducing high-concurrency read times by 40%. Verified parallel execution, load balancing hints, and connection pooling.
- Created reusable IaC modules with AWS CDK for microservices, S3 policies, Lambda orchestration, improving consistency, modularity, and DevSecOps compliance. Applied tag policies, least privilege, resource limits, and SSM parameters.
- Automated batch workflows and compliance data pipelines using Spring Batch, Quartz Scheduler, and Step Functions, improving reporting accuracy and submission speed. Audited compliance logs, retry logic, dead-letter queues, and SLA thresholds.
 Technologies & Tools:
- Java 17, Spring Boot, Spring Cloud, Spring Security, Microservices, Hibernate, PostgreSQL, Angular 14+, TypeScript, RxJS, HTML5, Bootstrap, AWS (EC2, Lambda, RDS, DynamoDB, API Gateway, CDK, Step Functions, IAM, S3, SQS, SNS), Kubernetes (EKS), Docker, Kafka, Kafka Streams, SageMaker, Cypress, Jest, GraphQL, REST, WebSockets, Jenkins, GitHub Actions, Gradle, Terraform, SonarQube, Quartz Scheduler, Redis, ELK Stack, Grafana, Kibana, CloudWatch, JIRA, WebLogic, SOAP, NiFi.

Java Full Stack Developer

Feb 2019 – Aug 2020

State of Nebraska | Lincoln, Nebraska

At the *State of Nebraska Judicial Branch*, I led the development of a secure case management system to automate legal workflows, court scheduling, and inter-agency document exchange. Legacy systems were migrated to AWS using CDK and Kubernetes, enhancing scalability, enforcing IAM-based access controls, and streamlining case resolution across courts and state departments.

- Developed and maintained a web-based case management system using Java 8, Spring Boot, and Angular 11, automating legal document workflows, case tracking, and compliance reporting. Integrated RESTful APIs, PostgreSQL, and AWS services to enhance security, scalability, inter-departmental data exchange, and document version control.
- Designed and implemented a microservices architecture with Spring Boot, Spring Cloud, and Docker, enabling modular case management, reducing inter-service dependencies, and improving system scalability, security, and operational efficiency.
- Built and deployed scalable RDS instances with PostgreSQL, AWS CDK, and Oracle 12c, optimizing schema migrations, enhancing security, and ensuring high-availability case data storage for seamless legal operations across agencies.
- Developed and optimized RESTful APIs using Spring MVC and JAX-RS, streamlining secure data exchange between state departments while enhancing performance, scalability, and compliance with legal and operational standards.
- Developed a real-time data pipeline using Apache Kafka, ensuring reliable message brokering and asynchronous communication across state departments with low-latency streaming and guaranteed message delivery.
- Implemented secure authentication and access controls using OAuth2, LDAP, and Spring Security, enforcing state security policies, SSO integration, session token renewal, and user identity management.
- Developed event-driven data replication pipelines using AWS DMS, Change Data Capture (CDC), and AWS CDK, optimizing real-time case data synchronization and reducing retrieval latency by 35%, ensuring consistency.
- Designed and developed an Angular-based UI for case tracking and workflow visualization, enhancing legal case monitoring, user experience, productivity, responsiveness, and accelerating application processing speed.
- Automated legal document generation with Spring Batch and Quartz Scheduler, minimizing manual effort, accelerating case resolution, reducing errors, and ensuring compliance with state processing standards.
- Led the migration of the legacy case management system to AWS, utilizing AWS CDK and Terraform to automate infrastructure deployment, enhance security, enforce tagging, and ensure compliance with regulations.
- Optimized SQL query performance in PostgreSQL and Oracle 12c by implementing indexing strategies, query tuning, caching mechanisms, partitioning, and improving case-related data retrieval efficiency by 35%.
- Developed interactive Tableau dashboards, delivering real-time insights into caseload distribution, judicial efficiency, backlog reduction, predictive analytics, and enabling data-driven decision-making for stakeholders.
- Developed a rule-based engine using Drools, automating eligibility verification, improving processing accuracy, audit transparency, workflow clarity, and reducing incorrect case approvals during evaluations.
- Implemented CI/CD pipelines with Jenkins, GitHub Actions, and Maven, streamlining automated deployments, security scans, rollback mechanisms, minimizing downtime, and ensuring high availability of legal applications.
- Optimized AWS RDS performance by fine-tuning parameters and implementing real-time monitoring via AWS CDK, improving read/write throughput, failover recovery, replication efficiency, and analytics reporting.
- Containerized applications using Docker, Kubernetes, and Azure Kubernetes Service (AKS), optimizing resource utilization, scalability, pod health checks, and automated workload management for seamless case processing.
- Refactored legacy JSP-based components into a modern Angular 11 UI, enhancing maintainability, responsiveness, code readability, UX design, and reducing technical debt while improving front-end performance.
- Developed event-driven workflows using Apache Camel, JMS, AWS Lambda, Step Functions, and API Gateway, automating case processing, document management, retries, notifications, and improving inter-departmental coordination.
- Enhanced system logging and monitoring using Log4j, Prometheus, Grafana, and Splunk, enabling real-time anomaly detection, metrics visualization, proactive resolution, and ensuring compliance with state security policies.
- Led production troubleshooting efforts using Postman for API validation, JIRA for issue tracking, and NiFi workflows for automated data transformation, monitoring scripts, and seamless integration across government systems.
- Technologies & Tools:

Java 8, Spring Boot, Spring MVC, Microservices, RESTful APIs, Hibernate, JPA, PostgreSQL, Oracle 12c, Angular 11, Azure Container Services, Azure SQL, Apache Kafka, JMS, LDAP, OAuth2, Drools, Docker, Kubernetes, Jenkins, Maven, JIRA, Log4j2, Prometheus, Grafana, Apache Camel, NiFi, Tableau, Splunk, AWS CDK, AWS Lambda, AWS API Gateway, AWS DMS, Terraform, Change Data Capture (CDC), Quartz Scheduler, GitHub Actions, Postman, OpenSearch, Elastic Stack (ELK), AWS Glue, AWS S3.

Sr. Java Full Stack Developer

Humana | Louisville, KY

At *Humana Health Insurance*, I led the engineering of a scalable, HIPAA-compliant real-time claims processing platform to modernize legacy systems, improve adjudication efficiency, reduce fraudulent claims, and streamline provider interactions. The architecture was microservices-based, deployed across AWS, and integrated with Kafka, OAuth2, and Drools to drive automation and security.

• Developed a real-time claims processing engine using Java 8, Spring Boot, and Microservices, reducing adjudication time by 40%, with integration of distributed tracing (Zipkin), OpenTelemetry, and fault-tolerant Kafka consumers for accurate claims auditing and end-to-end data visibility.

- Designed a dynamic, rules-based adjudication platform using Drools, PostgreSQL, and JPA to validate claims against policy conditions, automate eligibility verification, CPT code mapping, and reduce dependency on manual intervention by 50%.
- Migrated a legacy SOAP-based claims infrastructure to Spring MVC and JAX-RS RESTful APIs, standardizing payload formats, enabling Swagger/OpenAPI documentation, and introducing JSON schema validation for downstream system integration.
- Created a robust provider portal using Angular 7, RxJS, and OAuth2 with role-based access, enabling physicians to submit claims, view real-time statuses, and download statements securely using JWT-authenticated sessions.
- Built a React.js + Redux customer support dashboard with Redux-Saga and local state caching, automating member queries, pre-authorization approvals, and claim dispute resolutions with enhanced responsiveness and traceability.
- Developed a React Native mobile application integrating biometric authentication, offline storage, claim status push notifications, and appointment booking, offering users 24/7 access to real-time insurance services and document upload.
- Leveraged AWS EC2, S3, Lambda, and DynamoDB to build HIPAA-compliant, cost-effective infrastructure for claim record processing, enabling encryption-at-rest, access logging, lifecycle rules, and IAM role segregation.
- Automated AWS RDS infrastructure provisioning using AWS CDK with backup policies, VPC peering, multi-AZ failover support, and IAM integration, enabling real-time backups and restoring high-availability claims processing.
- Implemented event-driven architecture using Apache Kafka, RabbitMQ, and AWS SNS, enabling asynchronous communication across microservices, real-time streaming, deduplication handling, and insurance policy status broadcasting.
- Designed CI/CD pipelines with Jenkins, GitHub Actions, Docker, and Kubernetes for blue-green deployments, security scan automation, approval gates, release tagging, and on-demand rollback of critical claim services.
- Configured AWS RDS PostgreSQL cross-region replication via AWS CDK, enhancing business continuity, failover response, and backup snapshot automation, with CloudWatch-based failover notifications and recovery validation.
- Engineered a Kafka-based real-time fraud detection system using stream joins, ML-trained scoring thresholds, and Redis caching to reduce fraudulent payouts by 30% and prioritize high-risk claims for manual review.
- Created ML-based claim validation services using AWS SageMaker pipelines, XGBoost models trained on historical denial patterns, integrated with RESTful scoring endpoints and Airflow-triggered batch retraining jobs.
- Developed a Lambda and SNS-powered claims alerting system with S3-based content templates, localized email support, and compliance logging to inform users of real-time claim approvals, denials, or required actions.
- Built an OAuth2-enabled federated identity access layer with token revocation, refresh support, session timeout enforcement, and PKCE flow for enhanced security across patient and provider portals.
- Engineered secure S3 + DynamoDB-based document storage with versioning, signed URL enforcement, SSE-S3 encryption, IAM permission boundaries, and document expiration workflows for HIPAA compliance.
- Led the cloud modernization effort, replacing monolithic claim processors with AWS CDK-provisioned microservices, saving 30% cost and improving service boot time, latency, and fault tolerance significantly.
- Built a reliable message-processing system with ActiveMQ using DLQs, durable topics, retry policies, and message deduplication logic for large-scale bulk premium adjustments and policy changes.
- Enhanced system observability using Log4j2, Prometheus, Grafana, and CloudWatch custom metrics, integrating health probes, latency heatmaps, backlog trend graphs, and real-time anomaly alerts for claim workflows.

• Technologies & Tools:

Java 8, Spring Boot, Spring MVC, Microservices, RESTful APIs, Hibernate, JPA, PostgreSQL, Oracle DB, React.js, Redux, Angular 7, React Native, Apache Kafka, RabbitMQ, ActiveMQ, AWS (EC2, S3, Lambda, DynamoDB, RDS, SNS, SQS, Step Functions, CDK), Kubernetes, Docker, Jenkins, GitHub Actions, Maven, JUnit, Mockito, OAuth2, Drools, Log4j2, Tableau, Prometheus, Grafana, Swagger, Zipkin, Airflow, SageMaker, Terraform, Redshift, Athena.

Sr. Java Developer

Kaiser Permanente | San Francisco, CA

At *Kaiser Permanente*, I contributed to the development of a secure patient data integration platform and built a scalable microservices architecture supporting HIPAA-compliant patient data exchange, clinical interoperability, and cross-departmental analytics within the organization.

- Developed a patient data integration platform using Java 8, Spring Boot, and REST APIs to enable seamless real-time exchange of insurance and clinical records across hospital systems, enhancing provider workflows.
- Designed a microservices architecture using Spring Cloud and Docker containers, increasing modularity and fault tolerance while supporting patient record versioning and scalable analytics for care coordination teams.
- Implemented Kafka and ActiveMQ-based real-time event pipelines for asynchronous patient data delivery between insurers, pharmacy systems, and diagnostic labs, improving responsiveness and system decoupling.
- Built secure RESTful services using Spring MVC and JAX-RS with Swagger documentation and JWT-based authentication, improving interoperability and patient record access across legacy and cloud-based systems.
- Tuned complex SQL queries and indexing strategies in Oracle DB to reduce EHR report retrieval time by 35%, accelerating doctor decision-making for lab orders and prescriptions.
- Developed web-based dashboards using Vue.js and AngularJS to track patient discharge trends, prescription approvals, and claims status in real-time, improving physician visibility across departments.
- Automated CI/CD workflows using Jenkins and Maven with SonarQube integration, reducing manual testing efforts and ensuring high-quality deployments for insurance validation and data ingestion services.

Feb 2017 – Dec 2018

- Deployed Kubernetes-managed microservices for patient record ingestion and processing using Helm charts, rolling updates, and log monitoring with health check probes.
- Implemented RBAC using Spring Security and LDAP, ensuring HIPAA-compliant role-based record access, token lifecycle enforcement, and detailed audit trail logging for compliance teams.

• Technologies & Tools:

Java 8, Spring Boot, Spring MVC, RESTful APIs, SOAP, Hibernate, JPA, Oracle DB, MySQL, AngularJS 1.x, Vue.js, Apache Kafka, ActiveMQ, Docker, Kubernetes, Jenkins, Maven, LDAP, JWT, Log4j, WebSphere, Tomcat, UNIX, JasperReports, Tableau.

Java Developer

Jun 2016 – Jan 2017

Assurant Inc | Miami, FL

At *Assurant Inc*, I worked on developing and modernizing the insurance policy management system, streamlining policyholder data workflows, integrating fraud detection, and migrating monolith services into a fault-tolerant microservices architecture.

- Built policy management module using Spring Boot 1.4.1, Hibernate, and REST APIs, improving document workflows and ensuring effective processing of high-volume claims with secure design and scalable microservice components.
- Implemented microservices for claims adjudication using Spring Cloud, decoupling critical components and reducing dependency on legacy services, ensuring easier rollback, auto-scaling, fault isolation, and high availability support.
- Integrated Amazon Route 53 for global DNS routing, enabling faster regional failover and improving latency for geographically dispersed insurance policy services using low-latency request routing policies.
- Engineered fraud detection workflows with Kafka-based anomaly detectors, identifying suspicious claims in real-time using event correlation, aggregation, Redis tracking, and asynchronous stream processing logic.
- Created AngularJS dashboard with real-time charts, WebSocket data push, and auto-refresh intervals for insurance agents to monitor daily claim loads, alerts, and instant policy status notifications.
- Tuned MySQL and Oracle database performance by indexing policy tables, adding parallel execution hints, and rewriting expensive joins to improve data transaction speeds and reduce lock contention.
- Secured policy services with LDAP-based Spring Security roles, implementing audit logging, login throttling, session expiration, secure password hashing, and user scope checks aligned with PII compliance.
- Designed SOAP services compatible with legacy underwriting systems and exposed REST wrappers using Spring MVC for integration with newer React.js-based UIs and third-party insurance systems.
- Automated end-to-end deployments with Jenkins, Maven, and SVN hooks to ensure traceable version releases, Slack-based build notifications, rollback workflows, and enhanced observability using Log4j with Spring AOP for contextual trace logs, uptime metrics, and SLA monitoring.

• Technologies & Tools:

Java 8, Spring Boot 1.4.1, Hibernate, Spring MVC, RESTful APIs, SOAP, MySQL, Oracle DB, AngularJS 1.x, Amazon Route 53, Kafka, LDAP, Jenkins, Maven, Log4j, WebSphere, Tomcat, OpenShift, SVN.

Java Developer

Syneos Health | Jersey City, NJ

At *Syneos Health*, I developed a secure healthcare claims system to manage patient records, streamline fraud detection workflows, and modernize claims processing by replacing SOAP-based infrastructure with RESTful APIs.

- Developed HIPAA-compliant patient data processing platform using Spring Boot, REST APIs, and Hibernate; integrated record validation layers to securely exchange EHR with insurance partners.
- Built Spring Batch processes to handle bulk claim ingestion and medical record validation workflows, reducing SLA breaches and increasing parallelism for high-volume claim batches.
- Created interactive claims dashboard using Angular 1.x, jQuery, and Node.js for real-time claim visibility, integrating with backend REST services to deliver an intuitive experience to insurers and providers.
- Tuned Oracle SQL queries and indexing strategies, reducing latency in rule-based fraud models and accelerating detection of duplicates and anomalies in the insurance pipeline.
- Implemented secure encrypted messaging using JMS, IBM MQ, and durable topics to facilitate reliable communication between providers and insurer claim validation systems.
- Migrated SOAP endpoints to JSON RESTful APIs, unifying internal and partner interfaces and enabling schema evolution with Swagger versioning support.
- Built a validation engine with Drools, integrating it with claims ingest logic to enforce policy eligibility rules, minimize human error, and reduce time-consuming manual verification workflows.
- Deployed Jenkins-Maven-based pipelines to streamline code integration, add unit/integration test automation, and enforce coverage thresholds using JaCoCo and SonarQube.
- Deployed Java apps across WebSphere, WebLogic, and Tomcat, ensuring compatibility and consistent performance through multi-container testing and load balancing with Apache HTTPD.
- Built Kafka + JMS-powered notification system for insurance agents to receive real-time alerts on claim approval, escalation, or fraud intervention; integrated push services and event filtering.

Technologies & Tools:

Java 8, Spring Boot, Spring MVC, Hibernate, Spring Batch, JMS, IBM MQ, Kafka, SOAP, REST APIs, Drools, SQL, Oracle, Azure SQL, Angular 1.x, Node.js, jQuery, Eclipse, Maven, Jenkins, JIRA, WebSphere, WebLogic, Tomcat, SVN, Linux.

Mar 2014 – May 2016

SentimentSphere – Enterprise-Grade Generative AI Sentiment Intelligence Platform

Python, TensorFlow, Flask, PostgreSQL, Power BI, AWS (Lambda, S3, EC2, CloudWatch, SNS)

- Designed and deployed an enterprise-scale sentiment analytics pipeline leveraging BERT and RoBERTa models, achieving 93% F1-score across 100K+ customer interactions for CX benchmarking across telecom and finance sectors.
- Built and secured Flask-based APIs deployed on AWS Lambda, integrated with real-time ETL pipelines to power instant feedback analysis, behavioral routing, and escalation paths in customer support frameworks.
- Engineered a normalized PostgreSQL opinion mining schema with indexed views, enabling longitudinal mood tracking, sector-wise sentiment deltas, and alert-based deviation monitoring at sub-80ms latency.
- Developed rich Power BI executive dashboards visualizing sentiment trends, heatmaps, and polarity variance, driving cross-functional campaign realignments and contributing to a 25% increase in satisfaction scores.
- Automated model retraining workflows with EC2, CloudWatch, and SNS pipelines, enabling anomaly detection, bias drift alerting, and sustained model freshness with zero manual intervention.

RoamSafe – Scalable Roadside Assistance and Dispatch System

Java, Spring Boot, React.js, PostgreSQL, Docker, RabbitMQ, Google Maps API, AWS EC2, NGINX

- Engineered a robust microservices-based platform serving 10K+ users across multiple cities, dynamically matching stranded vehicles with verified mechanics using location proximity, ETA algorithms, and geofencing logic.
- Developed responsive React. js front-end with Google Maps API integration for real-time vehicle tracking, arrival prediction, mechanic validation, and user alerts with JWT-secured sessions.
- Designed PostgreSQL-backed SLA workflows and RabbitMQ queues for job dispatch, timeout retries, and resilience against service failures during peak traffic and disaster recovery scenarios.
- Deployed containerized services with Docker, secured reverse proxy via NGINX, and achieved sub-200ms response latency under concurrent load via horizontally scalable EC2 instances.
- Delivered custom operations dashboards with PostgreSQL views for backend agents to monitor service compliance, trip states, and mechanic activity, reducing dispatch times by 42%.