**Solution Architecture Document — Safaricom Digital Customer Service Portal**

**1. Introduction**

This document outlines the overall system architecture, components, technologies, and integration points for the Digital Customer Service Portal designed for Safaricom.

**2. Objectives**

* Provide a scalable, secure, and maintainable architecture.
* Enable seamless integration with existing CRM and third-party services.
* Support multi-channel customer interactions with high availability.

**3. Architecture Overview**

**High-Level Architecture Components:**

| **Component** | **Description** | **Technologies/Tools** |
| --- | --- | --- |
| User Interface Layer | Web portal and mobile apps for customer interaction. | ReactJS, Flutter (mobile), PWA |
| API Gateway | Centralized API entry point to backend services. | Kong, AWS API Gateway |
| Backend Microservices | Modular services for ticketing, chat, user management, analytics. | Node.js, Spring Boot, Docker |
| AI Chatbot Service | AI-powered conversational engine. | Google Dialogflow, IBM Watson |
| CRM Integration | Interface with Salesforce CRM for customer data and ticket sync. | REST APIs, OAuth2 authentication |
| Messaging & Notifications | Real-time chat, SMS, WhatsApp messaging. | Twilio, Kafka, WebSockets |
| Database Layer | Persistent storage for tickets, user data, logs. | PostgreSQL, MongoDB |
| Analytics & Reporting | Dashboards for monitoring KPIs and support metrics. | Power BI, Grafana |
| Security Layer | Authentication, authorization, encryption, and monitoring. | OAuth 2.0, JWT, TLS, IAM |
| Cloud Infrastructure | Hosting and scalability environment. | AWS/Azure, Kubernetes, Docker |

**4. Architecture Diagram**

**5. Detailed Component Descriptions**

* **User Interface Layer:**  
  Responsive web portal built with ReactJS and native mobile apps using Flutter to ensure consistent customer experience across devices. Supports accessibility features compliant with WCAG 2.1.
* **API Gateway:**  
  Handles all incoming requests, performs routing, throttling, security checks, and analytics before forwarding to backend microservices.
* **Backend Microservices:**  
  Each core function is a separate microservice (Ticket Management, User Profile, Chat, Reporting) deployed in containers for scalability and resilience.
* **AI Chatbot Service:**  
  Integrates with third-party conversational AI providers for initial customer interactions, with fallback to live agents.
* **CRM Integration:**  
  Syncs customer data, ticket updates, and customer interaction history bidirectionally with Salesforce CRM through secured REST APIs.
* **Messaging & Notifications:**  
  Real-time communication enabled by WebSockets and event-driven message queues. SMS and WhatsApp integration via Twilio APIs for broader reach.
* **Database Layer:**  
  Relational DB (PostgreSQL) for transactional data; NoSQL DB (MongoDB) for unstructured chat logs and AI training data.
* **Analytics & Reporting:**  
  Real-time KPIs, trend analysis, and SLA tracking visualized through Power BI dashboards accessible to customer service managers.
* **Security Layer:**  
  End-to-end encryption with TLS 1.2+, OAuth 2.0 authentication with JWT tokens, role-based access control, and continuous monitoring for anomalies.
* **Cloud Infrastructure:**  
  Hosted on AWS/Azure cloud with Kubernetes orchestration for container management, auto-scaling based on load, and disaster recovery configurations.

**6. Non-Functional Architecture Considerations**

| **Aspect** | **Description** |
| --- | --- |
| Scalability | Microservices and containerization enable scaling to support peak loads. |
| Availability | Multi-region deployment with failover support to achieve 99.9% uptime. |
| Security | Compliance with industry standards (ISO 27001, GDPR). Regular penetration testing planned. |
| Maintainability | Modular design facilitates easier updates and feature rollouts. |
| Performance | Use of caching layers (Redis), optimized queries, and CDN for static assets. |

**7. Technology Stack Summary**

| **Layer** | **Technology/Tool** | **Purpose** |
| --- | --- | --- |
| Frontend | ReactJS, Flutter | Customer portals and apps |
| Backend | Node.js, Spring Boot | Business logic and APIs |
| AI Chatbot | Dialogflow, Watson | Conversational AI |
| Database | PostgreSQL, MongoDB | Data storage |
| Messaging | Kafka, Twilio | Real-time communication |
| Cloud | AWS/Azure, Kubernetes | Hosting, orchestration |
| Security | OAuth 2.0, JWT, TLS | Authentication and encryption |
| Analytics | Power BI, Grafana | Reporting and monitoring |

**8. Integration Points**

| **System** | **Type** | **Description** |
| --- | --- | --- |
| Salesforce CRM | REST API | Customer and ticket data synchronization. |
| Twilio API | Messaging API | SMS and WhatsApp communication. |
| Identity Provider | OAuth 2.0 | Single Sign-On and user authentication. |

**9. Assumptions and Constraints**

* CRM APIs are stable and provide required data endpoints.
* Vendor APIs for messaging services maintain SLA.
* Cloud infrastructure is provisioned timely with required resources.
* Compliance audits and security reviews may introduce deployment delays.

**10. Approval**

| **Name** | **Role** | **Signature** | **Date** |
| --- | --- | --- | --- |
| Project Sponsor | CIO |  |  |
| Solution Architect | IT Lead |  |  |