Assessment Overview: Design a comprehensive data architecture for a field service management company to efficiently capture, process, and serve various data types to an AI/ML team. Focus on system design rather than implementation details.

Part 1: System Design Challenge (60% of assessment)

Scenario:

SmartField Inc. is a field service management company with the following data requirements:

1. **Data Sources:**
    - Operational database (SQL Server) containing customer records, work orders, and service history
    - IoT devices installed at customer sites sending telemetry data (temperature, pressure, usage metrics)
    - Mobile app used by field technicians capturing location data, photos of equipment, and service notes
    - CRM system with sales and marketing data
    - Customer support ticketing system
2. **Business Requirements:**
    - Enable predictive maintenance models to forecast equipment failures before they occur
    - Provide data for a recommendation system to optimize technician scheduling and routing
    - Support a customer churn prediction model
    - Train a computer vision model to detect equipment defects from images
    - Feed data into an LLM-based assistant for technicians

Deliverables:

1. **Architecture Diagram:**
    - Design a modern data lakehouse architecture using any cloud services
    - Specify data flow from source systems to consumption layers

- Implement medallion architecture (bronze/silver/gold)
- Include batch and streaming data paths

2. **Data Modeling Document:**
    - Propose key data models for the silver and gold layers
    - Define the structure for at least 3 critical data entities
    - Specify partitioning strategies and optimization techniques
3. **Data Governance Strategy:**
    - Outline approach for data quality, lineage tracking, and metadata management
    - Propose governance controls for sensitive customer data
    - Describe monitoring and alerting considerations
4. **ML/AI Data Serving Strategy:**
    - Detail how data will be prepared and served to ML/AI models
    - Address feature store requirements
    - Explain approach for handling both structured and unstructured data for model training

Part 2: Mini Implementation (40% of assessment)

Implement a small proof-of-concept to demonstrate your technical skills using open-source tools:

Task:

Using a public dataset that mimics field service operations, implement a simplified version of your proposed architecture:

Dataset Options:

- NYC 311 Service Requests Dataset (service tickets, locations, resolution times)
- Kaggle: Predictive Maintenance Dataset (machine sensor readings and failures)

Requirements:

1. **Data Pipeline:**

- Create a simple data pipeline using Apache Spark (PySpark)
- Implement bronze and silver layer transformations using Delta Lake format
- Demonstrate proper data quality validation between layers

2. **Feature Engineering:**
   - Create a notebook that transforms the processed data into features suitable for an ML model
   - Include both time-series features and categorical features
3. **Simple API:**
   - Implement a basic API endpoint (using FastAPI or Flask) that would serve data to an ML model
   - Include Swagger documentation for the API

Assessment Submission Guidelines

Please submit the following:

1. Architecture diagrams (as PDF or images)
2. Design documentation (PDF or Markdown)
3. Code for the mini-implementation (GitHub repository or zip file)
4. README explaining how to run your implementation locally
5. Brief presentation slides (5-10 slides) summarizing your approach and design decisions

Note: While the implementation portion requires working code, we're primarily evaluating your architectural thinking, design decisions, and approach to solving data engineering challenges for AI/ML use cases in a field service context.