AI-Powered ELCAD Drawing QA Assistant Proposal for TenneT

AI-Powered ELCAD Drawing QA Assistant Proposal for TenneT

# 1. Background & Challenge TenneT engineers must review hundreds of ELCAD drawing pages per substation project. Manual QA leads to missed errors, inconsistent documentation, and delays in contractor coordination. The increasing complexity of electrical layouts demands a scalable, intelligent QA solution.

# 2. Use Case Based Approach & TCS Proposition

Use Case 1: Document Error Discovery  
- Automatically detect missing labels, misclassified symbols, and layout mismatches.  
- TCS Proposition: Deploy a hybrid CV-NLP model using OpenCV + GPT-4 to parse layouts and annotate errors on each page.

Use Case 2: Pattern Detection & Repetition Highlighting  
- Identify errors repeated across multiple pages (e.g., missing relay labels).  
- TCS Proposition: Implement cross-page vector search to cluster repeated QA violations.

Use Case 3: Contractor Notification Workflow  
- Auto-detect most common issue and send alert to relevant contractor.  
- TCS Proposition: Integrate a simulated email trigger with TenneT's workflow or ticketing system.

Use Case 4: Combined PDF Report Generation  
- Generate a PDF summarizing page-by-page findings for audit and sign-off.  
- TCS Proposition: Embed a PDF engine to produce downloadable reports for compliance.

# 3. Success Criteria - >=95% accuracy in detecting QA violations. - Reduction in average QA review time by 60%. - Automated reports accepted by TenneT's audit team. - >90% contractor response rate to error alerts.

# 4. How AI/ML Will Be Used (Technical Overview) - Layout Analysis: PyMuPDF extracts each page as image. - Visual Detection: OpenCV/YOLOv8 for symbol matching and classification. - Symbol Validation: CLIP or CNN models verify spatial context. - NLP Reasoning: GPT-4 reviews layout rules and logs structured QA commentary. - Report Engine: FPDF module compiles structured findings. - Optional FAISS vector search clusters repeated issues across pages/projects.

# 5. Solution Architecture Diagram Components: - PDF Input - PyMuPDF Page Split - OpenCV / YOLO Detection - CLIP Symbol Context Validator - GPT-4 for QA Rules & Explanation - Report Generator (FPDF) - Optional: FAISS for Error Clustering - Contractor Alert Trigger (Email/Workflow)

# 6. TCS Expectations from TenneT - Access to a diverse library of ELCAD samples (annotated and clean). - Defined set of QA rules and examples of contractor error feedback. - Approval to test in pilot substation project.

# 7. Business Benefits - Faster QA cycles, reducing project delays. - Improved quality of documentation sent to regulatory auditors. - Standardized contractor feedback loop. - Scalable AI-first platform for electrical design QA.

# 8. TCS Practitioner View & Best Practices - Begin with assisted QA: show AI-detected errors, but let engineers verify. - Use error clustering to prioritize systematic issues. - Maintain audit logs and confidence scores per detection. - Implement feedback loop from engineers to retrain models.