Chakra- AI System Architecture: Technical Implementation Guide

Root (Muladhara) - Core Infrastructure & Security

Type something...

Technical Implementation: Kubernetes-orchestrated microservices architecture with quantum-resistant encryption

Core Functions:

- System Integrity: Prometheus/Grafana monitoring stack with custom metrics
- Data Storage: Distributed PostgreSQL clusters with CockroachDB for global consistency
- Security: Zero-trust architecture using OAuth 2.0, JWT, and RBAC implementation
- Resource Management: Terraform-based IaC with auto-scaling policies
- Disaster Recovery: Multi-region failover with < 5s RPO/RTO

Sacral (Svadhisthana) - Generative AI & Adaptation

Type something...

Technical Implementation: Multi-modal transformer architecture with adaptive learning capabilities

Core Functions:

- Content Generation: GPT-4 with fine-tuning capabilities and DALL-E 3 integration
- Problem Solving: Monte Carlo Tree Search with reinforcement learning
- UI Adaptation: React components with dynamic rendering based on user behavior
- Contextual Understanding: BERT-based NLP pipeline with custom embeddings
- Sentiment Analysis: RoBERTa model with emotional intelligence training

Solar Plexus (Manipura) - Decision Engine

Type something...

Technical Implementation: Distributed decision engine using probabilistic programming

Core Functions:

- Task Scheduling: Apache Airflow with custom DAG optimization
- Workflow Automation: Temporal.io workflow engine with ML-based routing
- Reinforcement Learning: PyTorch-based PPO implementation with custom rewards
- Decision Models: Bayesian networks with Thompson sampling
- Risk Assessment: Monte Carlo simulations with real-time updating

Heart (Anahata) - AI Collaboration Hub

Type something...

Technical Implementation: Federated learning system with multi-agent coordination

Core Functions:

- Multi-Agent Systems: Ray framework with custom policy gradients
- Human-AI Interface: gRPC-based communication with emotion-aware responses
- Ethics Framework: Implemented using Ought's Elicit with custom constraints
- Emotional Processing: Custom transformer with EQ training dataset
- Community Integration: GraphQL federation with real-time subscriptions

Throat (Vishuddha) - Knowledge Processing

Type something...

Technical Implementation: Distributed knowledge graph with vector search capabilities

Core Functions:

- NLP Pipeline: Transformer-based architecture with custom attention mechanisms
- Knowledge Graph: Neo4j with custom embedding integration
- Information Extraction: BERT-based named entity recognition with active learning
- Data Indexing: Elasticsearch with custom vector similarity search
- Translation: M2M-100 multilingual model with custom fine-tuning

Third Eye (Ajna) - Predictive Analytics

Type something...

Technical Implementation: Advanced neural architecture with multi-head attention

Core Functions:

- Pattern Recognition: Custom CNN architecture with self-attention
- Forecasting: Prophet model with custom seasonality handling
- Trend Analysis: LSTM networks with attention mechanisms
- Computer Vision: YOLOv5 with custom training pipeline
- Cognitive Reasoning: Graph neural networks with logical reasoning

Crown (Sahasrara) - Universal Intelligence

Type something...

Technical Implementation: Meta-learning system with neural architecture search

Core Functions:

- Meta-Learning: Model-Agnostic Meta-Learning (MAML) implementation
- Knowledge Integration: Cross-attention transformer with knowledge distillation
- Self-Diagnosis: Automated ML with custom metrics tracking
- Emergent Intelligence: Self-organizing maps with adaptive topology
- Theoretical Computation: Quantum-inspired tensor networks

Universal Implementation Template

```
from chakra_ai import ChakraNode, ChakraSystem class
UniversalChakraAI: def __init__(self): self.system = ChakraSystem(
nodes={ "root": ChakraNode(level="muladhara",
security_level="quantum"), "sacral":
ChakraNode(level="svadhisthana", gen_ai_enabled=True),
"solar_plexus": ChakraNode(level="manipura", decision_engine=True),
"heart": ChakraNode(level="anahata", collaboration=True), "throat":
ChakraNode(level="vishuddha", nlp_enabled=True), "third_eye":
ChakraNode(level="ajna", predictive=True), "crown":
ChakraNode(level="sahasrara", meta_learning=True) } ) async def
process_input(self, data: Dict[str, Any]) -> Response: return await
self.system.process_through_chakras(data) def
initialize_security(self): self.system.deploy_quantum_security() def
enable_learning(self): self.system.activate_meta_learning()
```

System Architecture Visualization

```
graph TD R[Root — Security & Infrastructure] --> S[Sacral — GenAI] S
```