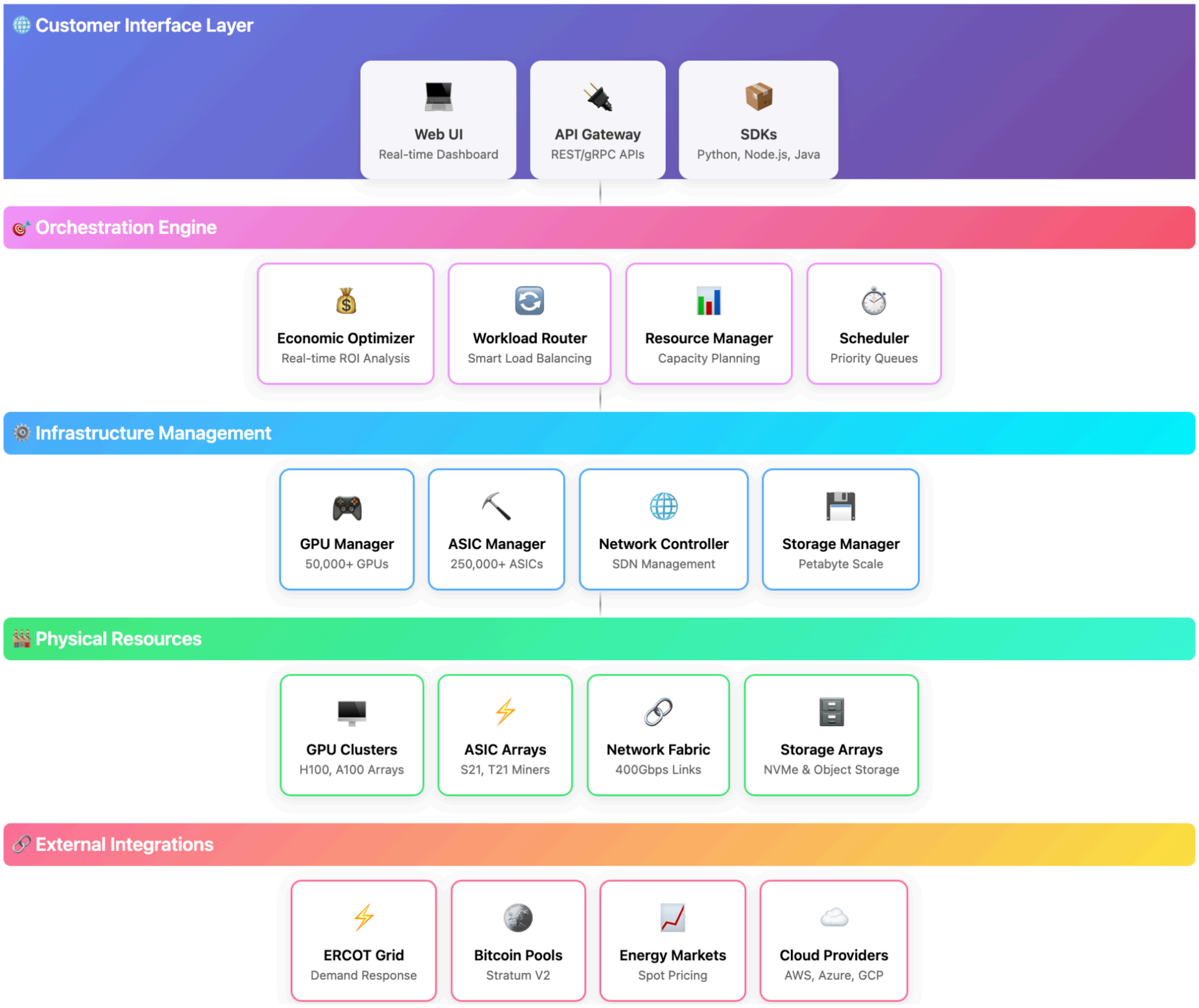
Software Flow Visual



Customer Interface Layer

Web UI
Real-time Dashboard

API Gateway
REST/gRPC APIs

SDKs
Python, Node.js, Java

Orchestration Engine

Economic Optimizer
Real-time ROI Analysis

Workload Router
Smart Load Balancing

Resource Manager
Capacity Planning

Scheduler
Priority Queues

Infrastructure Management

GPU Manager
50,000+ GPUs

ASIC Manager
250,000+ ASICs

Network Controller
SDN Management

Storage Manager
Petabyte Scale

Physical Resources

GPU Clusters
H100, A100 Arrays

ASIC Arrays
S21, T21 Miners

Network Fabric
400Gbps Links

Storage Arrays
NVMe & Object Storage

External Integrations

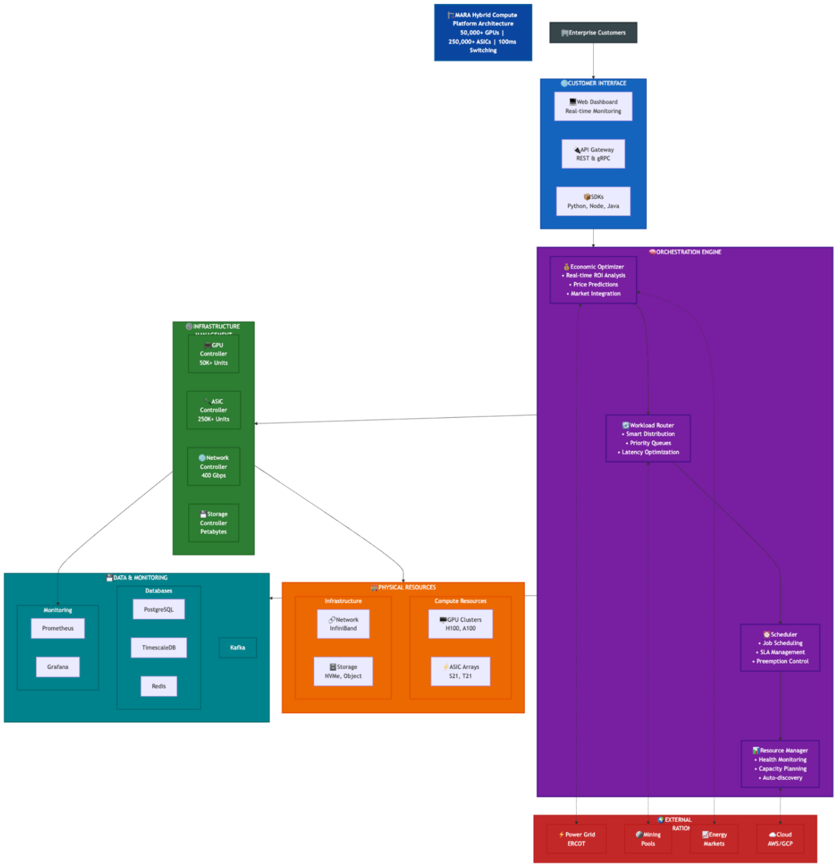
ERCOT Grid
Demand Response

Bitcoin Pools
Stratum V2

Energy Markets
Spot Pricing

Cloud Providers
AWS, Azure, GCP

Mermaid Chart -



Mermaid Graph Code -

```
flowchart TB
    %% ====== HEADER ======
    TITLE[" 🏗️ MARA Hybrid Compute Platform Architecture<br/>50,000+ GPUs | 250,000+ ASICs | 100ms Switching "]

    %% ====== USER ENTRY ======
    USER[" 🏢 Enterprise Customers "]

    USER --> INTERFACE

    %% ====== LAYER 1: INTERFACE ======
    subgraph INTERFACE[" 🌐 CUSTOMER INTERFACE "]
        direction LR
        WEB[" 💻 Web Dashboard<br/>Real-time Monitoring "]
        API[" 🔌 API Gateway<br/>REST & gRPC "]
        SDK[" 📦 SDKs<br/>Python, Node, Java "]
    end

    INTERFACE --> ENGINE

    %% ====== LAYER 2: BRAIN ======
    subgraph ENGINE[" 🧠 ORCHESTRATION ENGINE "]
        direction TB

        ECO[" 💰 Economic Optimizer<br/>• Real-time ROI Analysis<br/>• Price Predictions<br/>• Market Integration "]

        ECO --> ROUTE

        ROUTE[" 🔄 Workload Router<br/>• Smart Distribution<br/>• Priority Queues<br/>• Latency Optimization "]

        ROUTE --> SCHED

        SCHED[" ⏰ Scheduler<br/>• Job Scheduling<br/>• SLA Management<br/>• Preemption Control "]

        SCHED --> RESMGR

        RESMGR[" 📊 Resource Manager<br/>• Health Monitoring<br/>• Capacity Planning<br/>• Auto-discovery "]
    end

    ENGINE --> CONTROLLERS

    %% ====== LAYER 3: CONTROLLERS ======
    subgraph CONTROLLERS[" ⚙️ INFRASTRUCTURE MANAGEMENT "]
        direction LR

        GPU_CTRL[" 🎮 GPU<br/>Controller<br/>50K+ Units "]
        ASIC_CTRL[" ⛏️ ASIC<br/>Controller<br/>250K+ Units "]
        NET_CTRL[" 🌐 Network<br/>Controller<br/>400 Gbps "]
        STOR_CTRL[" 💾 Storage<br/>Controller<br/>Petabytes "]
    end
```

```
CONTROLLERS --> PHYSICAL

%% ====== LAYER 4: HARDWARE ======
subgraph PHYSICAL[" 🏭 PHYSICAL RESOURCES "]
    direction TB

    subgraph COMPUTE[" Compute Resources "]
        direction LR
        GPU_FARM[" 🖥️ GPU Clusters<br/>H100, A100 "]
        ASIC_FARM[" ⚡ ASIC Arrays<br/>S21, T21 "]
    end

    subgraph INFRA[" Infrastructure "]
        direction LR
        NETWORK[" 🔗 Network<br/>InfiniBand "]
        STORAGE[" 🗄️ Storage<br/>NVMe, Object "]
    end
end

%% ====== EXTERNAL SYSTEMS ======
subgraph EXTERNAL[" 🌍 EXTERNAL INTEGRATIONS "]
    direction LR
    GRID[" ⚡ Power Grid<br/>ERCOT "]
    POOLS[" 🪙 Mining<br/>Pools "]
    MARKET[" 📈 Energy<br/>Markets "]
    CLOUD[" ☁️ Cloud<br/>AWS/GCP "]
end

%% ====== DATA LAYER ======
subgraph DATA[" 💾 DATA & MONITORING "]
    direction LR

    subgraph DB[" Databases "]
        PG[" PostgreSQL "]
        TS[" TimescaleDB "]
        RD[" Redis "]
    end

    subgraph MON[" Monitoring "]
        PROM[" Prometheus "]
        GRAF[" Grafana "]
    end
```

```
       KAFKA[" Kafka "]
   end

   %% ====== KEY CONNECTIONS ======
   ECO <-.-> MARKET
   ECO <-.-> GRID
   ROUTE <-.-> POOLS
   RESMGR <-.-> CLOUD

   ENGINE --> DATA
   CONTROLLERS --> MON

   %% ====== STYLING ======
   classDef headerStyle
fill:#0d47a1,stroke:#01579b,color:#fff,stroke-width:4px,font-weight:bold
   classDef userStyle fill:#37474f,stroke:#263238,color:#fff,stroke-width:3px
   classDef interfaceStyle fill:#1565c0,stroke:#0d47a1,color:#fff,stroke-width:2px
   classDef engineStyle fill:#7b1fa2,stroke:#4a148c,color:#fff,stroke-width:3px
   classDef controlStyle fill:#2e7d32,stroke:#1b5e20,color:#fff,stroke-width:2px
   classDef physicalStyle fill:#ef6c00,stroke:#e65100,color:#fff,stroke-width:3px
   classDef externalStyle fill:#c62828,stroke:#b71c1c,color:#fff,stroke-width:2px
   classDef dataStyle fill:#00838f,stroke:#006064,color:#fff,stroke-width:2px

   class TITLE headerStyle
   class USER userStyle
   class INTERFACE interfaceStyle
   class ENGINE,ECO,ROUTE,SCHED,RESMGR engineStyle
   class CONTROLLERS,GPU_CTRL,ASIC_CTRL,NET_CTRL,STOR_CTRL controlStyle
   class PHYSICAL,COMPUTE,INFRA physicalStyle
   class EXTERNAL,GRID,POOLS,MARKET,CLOUD externalStyle
   class DATA,DB,MON,KAFKA dataStyle
```