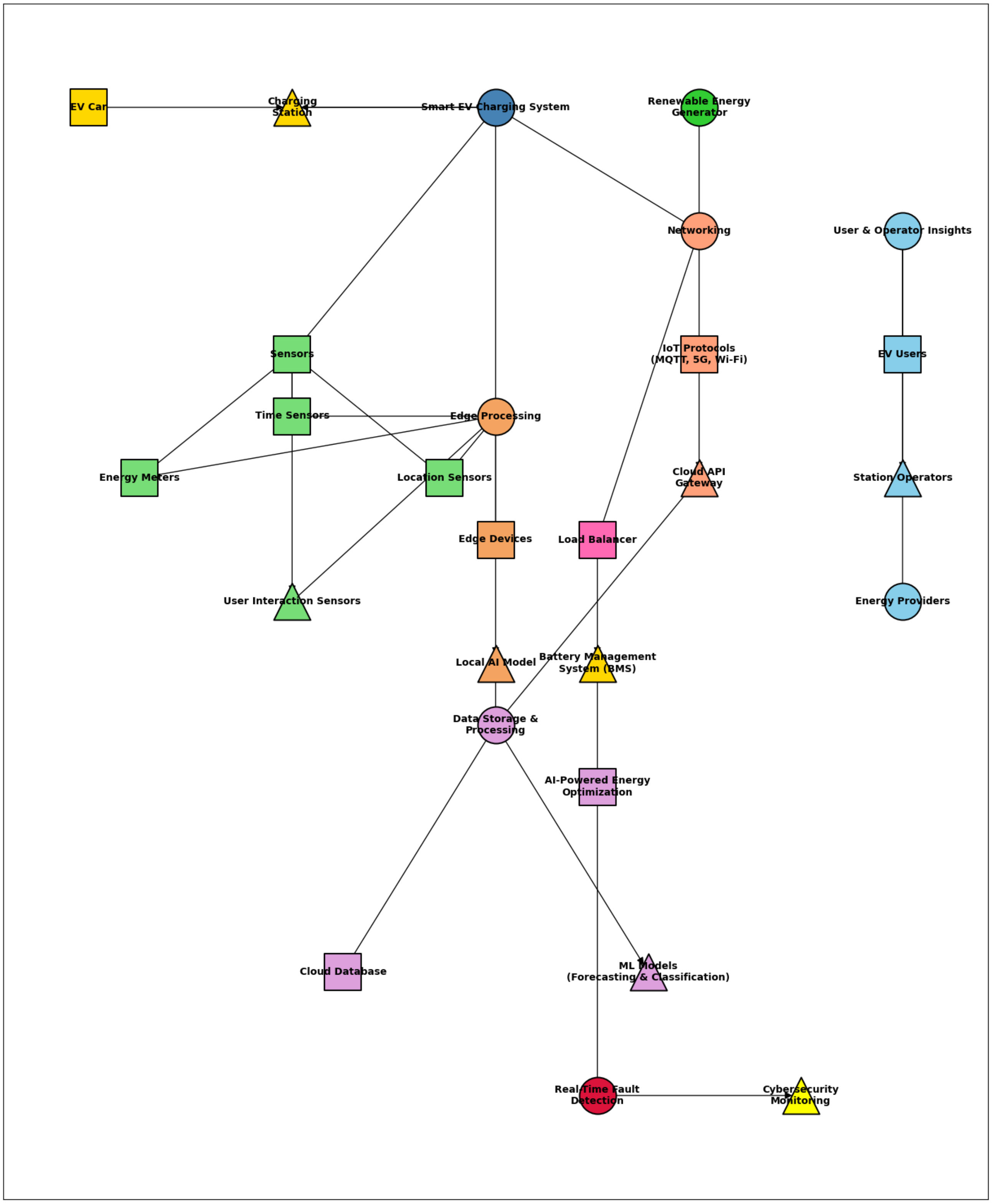
In [150…

```
# IoT System Design Diagram

import matplotlib.pyplot as plt
import networkx as nx

# Create a directed graph for the IoT system design
G = nx.DiGraph()

# Define nodes with positions, shapes, and colors for better visualization
nodes = {
    "Smart EV Charging System": {"pos": (5, 12), "shape": "o", "color": "#46

    # Sensors Layer
    "Sensors": {"pos": (3, 10), "shape": "s", "color": "#77DD77"},
    "Energy Meters": {"pos": (1.5, 9), "shape": "s", "color": "#77DD77"},
    "Time Sensors": {"pos": (3, 9.5), "shape": "s", "color": "#77DD77"},
    "Location Sensors": {"pos": (4.5, 9), "shape": "s", "color": "#77DD77"},
    "User Interaction Sensors": {"pos": (3, 8), "shape": "^", "color": "#77D

    # Edge Processing Layer
    "Edge Processing": {"pos": (5, 9.5), "shape": "o", "color": "#F4A460"},
    "Edge Devices": {"pos": (5, 8.5), "shape": "s", "color": "#F4A460"},
    "Local AI Model": {"pos": (5, 7.5), "shape": "^", "color": "#F4A460"},

    # Networking Layer
    "Networking": {"pos": (7, 11), "shape": "o", "color": "#FFA07A"},
    "IoT Protocols\n(MQTT, 5G, Wi-Fi)": {"pos": (7, 10), "shape": "s", "colo
    "Cloud API\nGateway": {"pos": (7, 9), "shape": "^", "color": "#FFA07A"},

    # Data Storage & Machine Learning
    "Data Storage &\nProcessing": {"pos": (5, 7), "shape": "o", "color": "#D
    "Cloud Database": {"pos": (3.5, 5), "shape": "s", "color": "#DDA0DD"},
    "ML Models\n(Forecasting & Classification)": {"pos": (6.5, 5), "shape":

    # User & Operator Insights
    "User & Operator Insights": {"pos": (9, 11), "shape": "o", "color": "#87
    "EV Users": {"pos": (9, 10), "shape": "s", "color": "#87CEEB"},
    "Station Operators": {"pos": (9, 9), "shape": "^", "color": "#87CEEB"},
    "Energy Providers": {"pos": (9, 8), "shape": "o", "color": "#87CEEB"},

    # Additional Components
    "EV Car": {"pos": (1, 12), "shape": "s", "color": "#FFD700"},
    "Charging\nStation": {"pos": (3, 12), "shape": "^", "color": "#FFD700"},
    "Renewable Energy\nGenerator": {"pos": (7, 12), "shape": "o", "color": "

    # Additional Nodes from Second Diagram
    "Load Balancer": {"pos": (6, 8.5), "shape": "s", "color": "#FF69B4"},
    "Battery Management\nSystem (BMS)": {"pos": (6, 7.5), "shape": "^", "col
    "AI-Powered Energy\nOptimization": {"pos": (6, 6.5), "shape": "s", "colo
    "Real-Time Fault\nDetection": {"pos": (6, 4), "shape": "o", "color": "#D
    "Cybersecurity\nMonitoring": {"pos": (8, 4), "shape": "^", "color": "#FF
}

# Define edges (connections between components)
edges = [
    ("Smart EV Charging System", "Sensors"),
    ("Smart EV Charging System", "Networking"),
    ("Smart EV Charging System", "Edge Processing"),
    ("Smart EV Charging System", "Charging\nStation"),

    ("Sensors", "Energy Meters"),
    ("Sensors", "Time Sensors"),
    ("Sensors", "Location Sensors"),
    ("Sensors", "User Interaction Sensors"),

    ("Energy Meters", "Edge Processing"),
    ("Time Sensors", "Edge Processing"),
    ("Location Sensors", "Edge Processing"),
    ("User Interaction Sensors", "Edge Processing"),

    ("Edge Processing", "Edge Devices"),
    ("Edge Processing", "Local AI Model"),
    ("Local AI Model", "Data Storage &\nProcessing"),

    ("Networking", "IoT Protocols\n(MQTT, 5G, Wi-Fi)"),
    ("IoT Protocols\n(MQTT, 5G, Wi-Fi)", "Cloud API\nGateway"),
    ("Cloud API\nGateway", "Data Storage &\nProcessing"),

    ("Data Storage &\nProcessing", "Cloud Database"),
    ("Data Storage &\nProcessing", "ML Models\n(Forecasting & Classificatior

    ("User & Operator Insights", "EV Users"),
    ("User & Operator Insights", "Station Operators"),
    ("User & Operator Insights", "Energy Providers"),

    ("EV Car", "Charging\nStation"),
    ("Charging\nStation", "Smart EV Charging System"),
    ("Renewable Energy\nGenerator", "Networking"),

    ("Networking", "Load Balancer"),
    ("Load Balancer", "Battery Management\nSystem (BMS)"),
    ("Battery Management\nSystem (BMS)", "AI-Powered Energy\nOptimization"),
    ("AI-Powered Energy\nOptimization", "Real-Time Fault\nDetection"),
    ("Real-Time Fault\nDetection", "Cybersecurity\nMonitoring"),
]

# Add nodes and edges to the graph
G.add_nodes_from(nodes.keys())
```

```
G.add_edges_from(edges)

# Draw the graph with formatted nodes
plt.figure(figsize=(18, 22))  # Increase height to enhance clarity
pos = {node: attributes["pos"] for node, attributes in nodes.items()}

# Draw edges first to keep them behind nodes
nx.draw_networkx_edges(G, pos, edge_color="black", arrows=True, arrowsize=15

# Draw nodes with different shapes
for node, attributes in nodes.items():
    x, y = attributes["pos"]
    if attributes["shape"] == "o":
        plt.scatter(x, y, s=1400, color=attributes["color"], edgecolors="bla
    elif attributes["shape"] == "s":
        plt.scatter(x, y, s=1400, color=attributes["color"], marker="s", edg
    elif attributes["shape"] == "^":
        plt.scatter(x, y, s=1400, color=attributes["color"], marker="^", edg

# Draw labels with improved spacing
nx.draw_networkx_labels(G, pos, font_size=10, font_weight="bold")

# Show the final refined IoT System Design Diagram
plt.show()
```


EV Car
Charging Station
Smart EV Charging System
Renewable Energy Generator
Networking
User & Operator Insights
Sensors
IoT Protocols (MQTT, 5G, Wi-Fi)
EV Users
Time Sensors
Edge Processing
Energy Meters
Location Sensors
Cloud API Gateway
Station Operators
Edge Devices
Load Balancer
User Interaction Sensors
Energy Providers
Local AI Model
Battery Management System (BMS)
Data Storage & Processing
AI-Powered Energy Optimization
Cloud Database
ML Models (Forecasting & Classification)
Real-Time Fault Detection
Cybersecurity Monitoring

```
In [ ]:
```