```
import requests
import json
import pandas as pd
from datetime import datetime

# 1. OpenWeatherMap API - Current Weather Data
def get_weather_data(lat, lon, api_key):
    url = f"https://api.openweathermap.org/data/2.5/weather?lat={lat}&lon={lon}&units=metric&appid={api_key}"
    response = requests.get(url)
    data = response.json()
    temperature = data['main']['temp']
    timestamp = datetime.utcfromtimestamp(data['dt']).isoformat() + 'Z'
    return {'type': 'weather', 'metric': 'temperature', 'value': temperature, 'timestamp': timestamp}

# 2. Alpha Vantage API - Stock Prices Data
def get_stock_data(symbol, api_key):
    url = f"https://www.alphavantage.co/query?function=TIME_SERIES_DAILY&symbol={symbol}&apikey={api_key}"
    response = requests.get(url)
    data = response.json()
    latest_date = list(data['Time Series (Daily)'].keys())[0]
    stock_price = float(data['Time Series (Daily)'][latest_date]['4. close'])
    timestamp = latest_date + 'T00:00:00Z'  # Align timestamp to UTC
    return {'type': 'stock', 'metric': 'asset_price', 'value': stock_price, 'timestamp': timestamp}

# 3. CoinGecko API - Cryptocurrency Data
def get_crypto_data(crypto_id):
    url = f"https://api.coingecko.com/api/v3/coins/{crypto_id}"
    headers = {"accept": "application/json"}
    response = requests.get(url, headers=headers)
    data = response.json()
    crypto_price = data['market_data']['current_price']['usd']
    timestamp = data['last_updated']
    return {'type': 'crypto', 'metric': 'asset_price', 'value': crypto_price, 'timestamp': timestamp}

# Pretty print for displaying results
def pretty_print(label, data):
    chars = 200
    print('-' * chars)
    print(label.upper())
    print('-' * chars)

    # Convert data to JSON serializable format (convert timestamps to string)
    if isinstance(data, list):
        for item in data:
            if 'timestamp' in item:
                item['timestamp'] = item['timestamp'].isoformat()  # Convert to ISO format string

    print(json.dumps(data, indent=4))
    print('-' * chars)

# ETL Pipeline
def run_etl_pipeline():
    openweathermap_key = 'b4e6c2f82b483b15c45a0ca314c1134a'
    alphavantage_key = '6WDE9VL2XGZNK0HM'

    # Extract data
    weather_data = get_weather_data(lat=51.5074, lon=-0.1278, api_key=openweathermap_key)  # Example: London
    stock_data = get_stock_data(symbol='AAPL', api_key=alphavantage_key)
    crypto_data = get_crypto_data(crypto_id='bitcoin')

    # Load into DataFrame
    df = pd.DataFrame([weather_data, stock_data, crypto_data])

    # Transformations
    df['value'] = pd.to_numeric(df['value'], errors='coerce')  # Ensure 'value' is numeric

    # Parse 'timestamp' to datetime with UTC handling
    df['timestamp'] = pd.to_datetime(df['timestamp'], utc=True, format='ISO8601')  # Parse timestamps with ISO8601 format

    # Display transformed data
    print("Transformed Data:")
    print(df)

    # Aggregation: Summarize data (daily) by metric
    daily_summary = df.groupby([df['timestamp'].dt.date, 'metric'])['value'].mean().reset_index()
```

```
    # Display aggregated data
    print("\nDaily Summary:")
    print(daily_summary)

    return df, daily_summary

# Run ETL pipeline
df, daily_summary = run_etl_pipeline()

# Display output
pretty_print("Extracted Data", df.to_dict(orient='records'))
pretty_print("Daily Summary", daily_summary.to_dict(orient='records'))
```