Presenting

HelioFlux

a submission for Luminous Techno-X by **TimE**

Piyush Ojha
Suryansh Srivastava

Develop a platform that optimizes energy consumption for households and businesses with solar panels, based on time-of-use (TOU) or time-of-day (TOD) electricity tariffs. The platform should help users maximize savings by shifting energy usage to periods of lower electricity costs and utilizing stored solar energy during peak pricing times

Live Energy Monitoring – Always in the Loop
Stay ahead of the energy game with real-time insights on every watt used!

The Hybrid Brain – Making Smarter Decisions
A genius mix of prediction and action – always learning, always optimizing.

HelioFlux

Your Intelligent Energy Management Solution

Energy Ninja – The Art of Smart Scheduling
Masters the art of low-tariff timing like a true stealthy energy warrior.

Solar Sentinel – Guarding Your Energy Source
The ultimate solar watchdog, ensuring no sunbeam goes to waste!

Alert Mode – Your Early Warning System
The first to know, so you're never caught off-guard by energy spikes.

Live Energy Monitoring - Always in the Loop

Our app collects and pre-processes real-time data from your **home, appliances, solar power systems, and Time-of-Use (TOU) electricity rates**, ensuring every detail is ready for intelligent decision-making.

Two sample datasets were generated to simulate real-time data for the prototype phase. These datasets will be replaced with live data during the application's development stage.

Time of Use Electricity Tariff Dataset

This dataset tracks the fluctuations in electricity tariffs throughout the day. This includes:

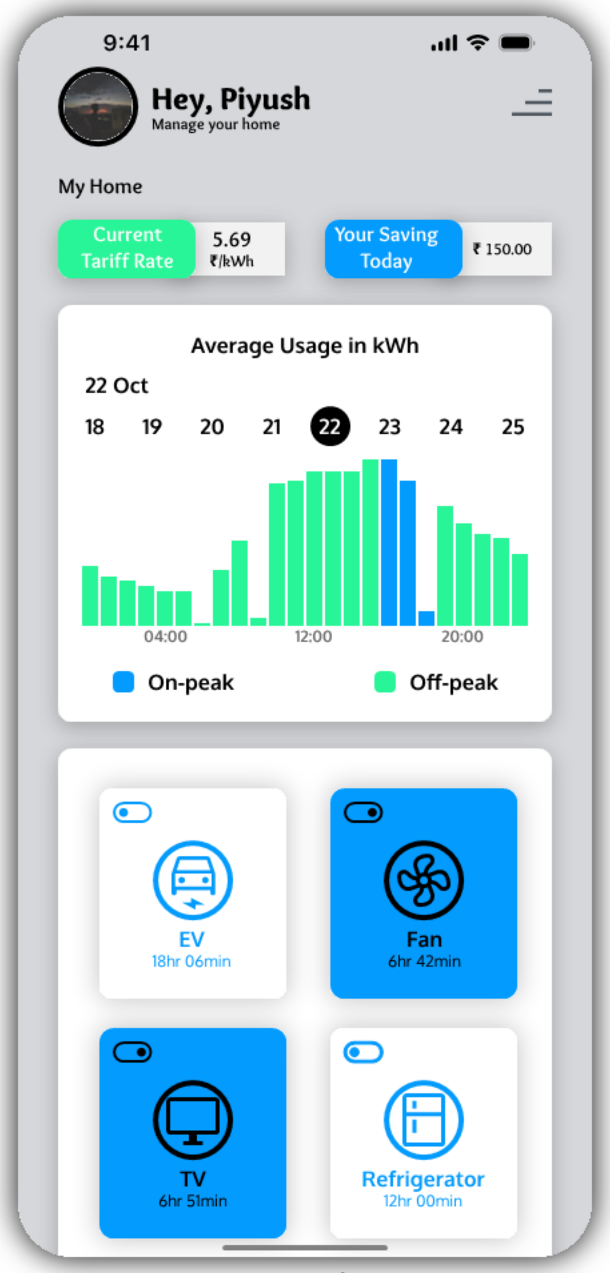
- **Timestamp** - Date and time when the tariff rate is applicable
- **Tariff_Rate (₹/kWh)** - The electricity rate at that time
- **Tariff_Type** - Specifies if it's a peak, off-peak, or standard period
- **TOU_Period_Start** - Start time of the specific time-of-use period
- **TOU_Period_End** - End time of the specific period
- **Weather_Conditions** - Include data on weather patterns if available (e.g., sunny, cloudy)
- **Grid_Load** - The total energy demand on the grid during the time period

Customer Daily Usage Dataset

This dataset tracks the usage pattern for the day. This includes:

- **Appliance_type:** The type of appliance (e.g., Washing Machine, Dishwasher, Air Conditioner, EV Charger).
- **Power_Rating (kWh):** The average power consumption of the appliance per hour.
- **Usage_Duration (hours):** Total hours the appliance is used daily (or monthly).
- **Usage_Time_Start:** Time of day when the appliance starts operating.
- **Usage_Time_End:** Time of day when the appliance finishes operating.
- **Days_of_Usage:** When the appliance is typically used (e.g., weekdays or weekends).
- **Seasonal_Variation:** Whether appliance usage varies by season (important for heating/cooling appliances).

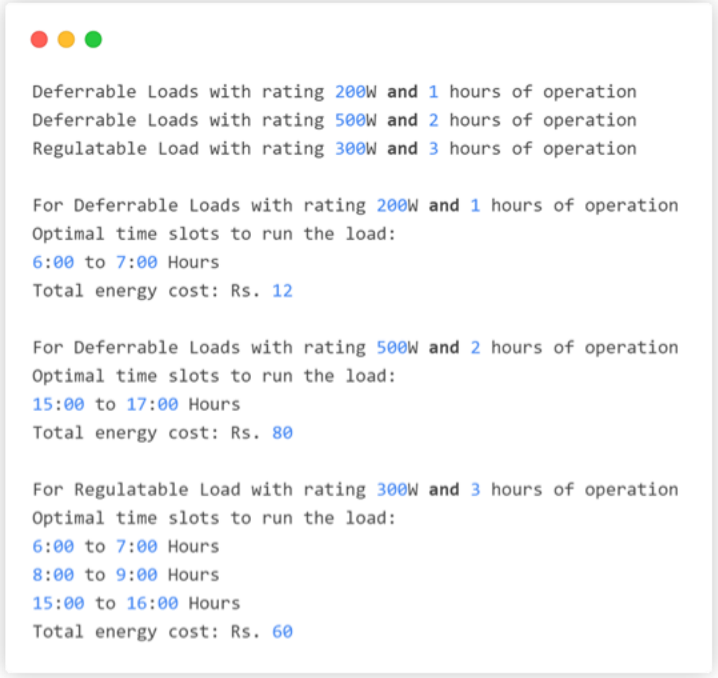
Cloud-based data processing will handle all computations, with only the **results being retrieved on users' devices**. Tariff rates will be fetched via an API.
Data on daily usage patterns will be gathered, analysed, and uploaded to the cloud server, allowing for continuous **refinement and optimization of the algorithm.**



Home Page of Our App

Energy Ninja - the Art of Smart Scheduling

Our system automates **appliance scheduling**, optimizing operation **based on electricity prices and solar availability**, maximizing savings without compromising convenience.

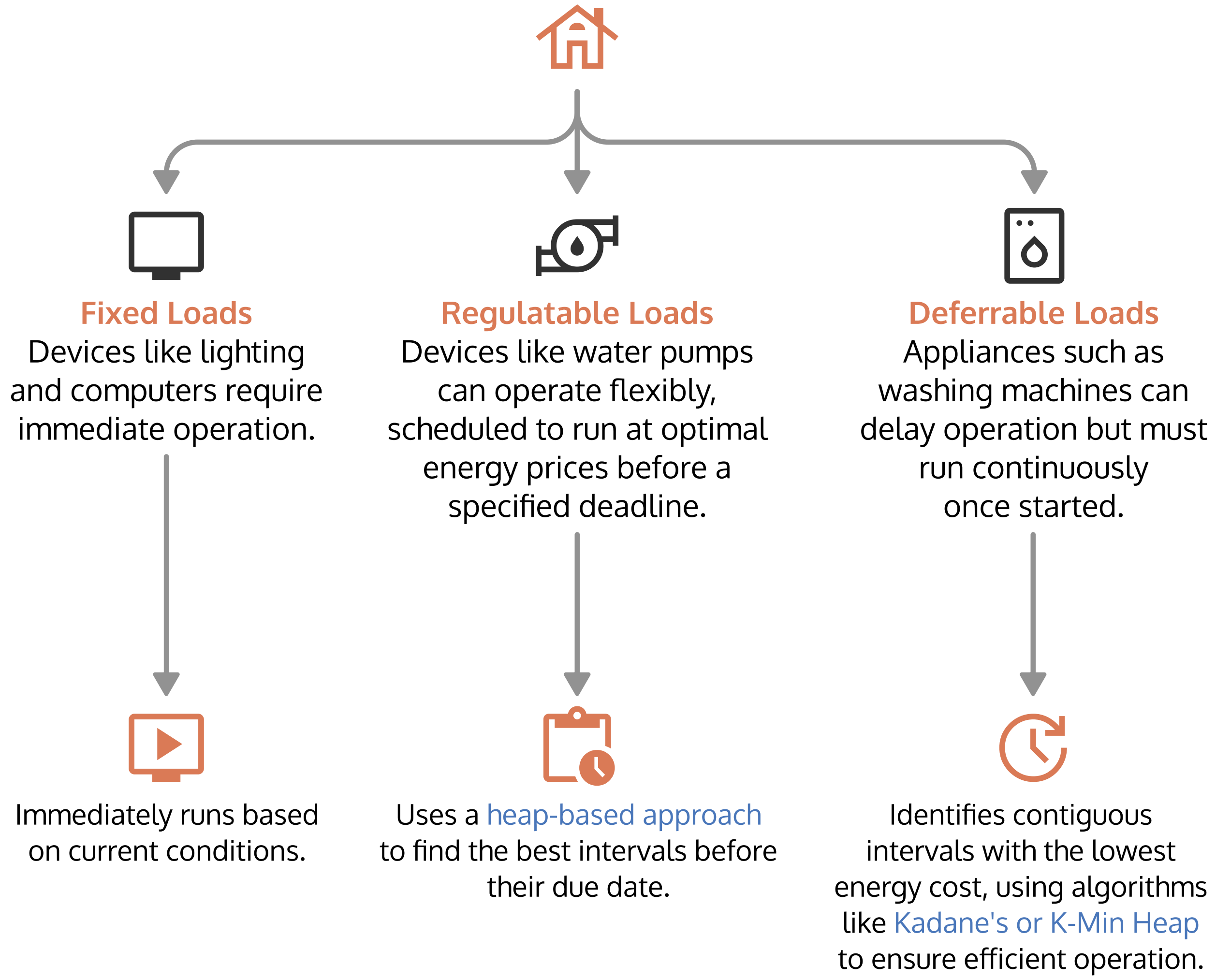


```
Deferrable Loads with rating 200W and 1 hours of operation
Deferrable Loads with rating 500W and 2 hours of operation
Regulatable Load with rating 300W and 3 hours of operation

For Deferrable Loads with rating 200W and 1 hours of operation
Optimal time slots to run the load:
6:00 to 7:00 Hours
Total energy cost: Rs. 12

For Deferrable Loads with rating 500W and 2 hours of operation
Optimal time slots to run the load:
15:00 to 17:00 Hours
Total energy cost: Rs. 80

For Regulatable Load with rating 300W and 3 hours of operation
Optimal time slots to run the load:
6:00 to 7:00 Hours
8:00 to 9:00 Hours
15:00 to 16:00 Hours
Total energy cost: Rs. 60
```

Scheduling Algorithm Result

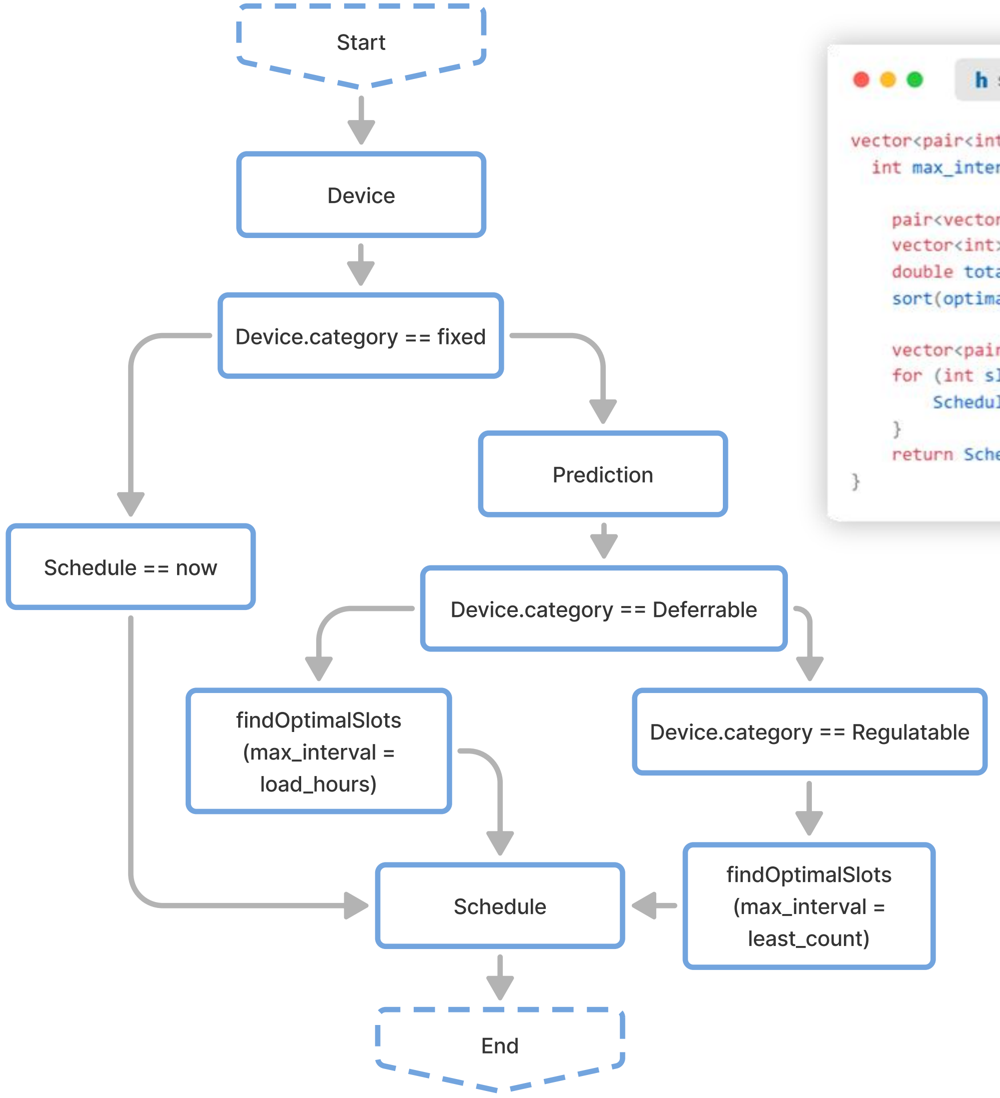
Results:

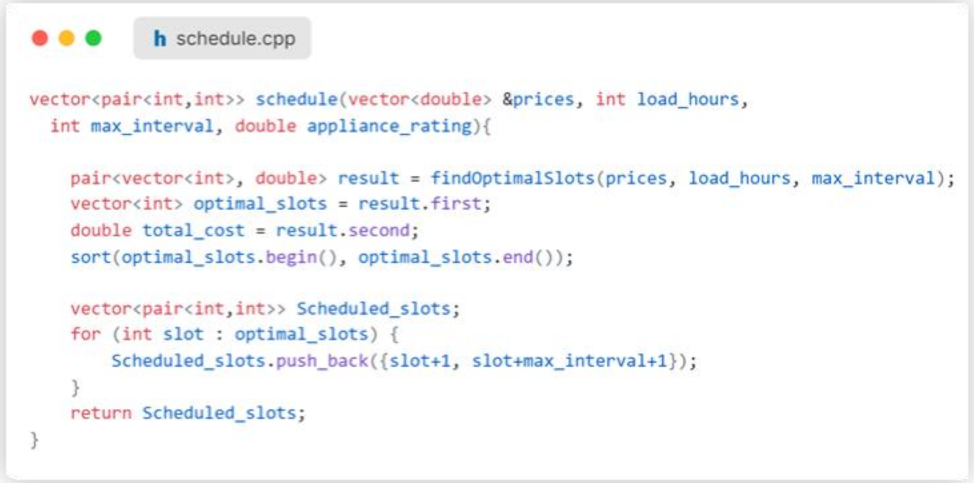
- **For Fixed Loads:** Schedules them for immediate operation.
- **For Deferrable Loads:** Identifies the optimal time interval and schedules them accordingly.
- **For Regulatable Loads:** Segments the total operation time, then selects and schedules the optimal intervals for each segment.


Fixed Loads
Devices like lighting and computers require immediate operation.
Regulatable Loads
Devices like water pumps can operate flexibly, scheduled to run at optimal energy prices before a specified deadline.
Deferrable Loads
Appliances such as washing machines can delay operation but must run continuously once started.
Immediately runs based on current conditions.
Uses a heap-based approach to find the best intervals before their due date.
Identifies contiguous intervals with the lowest energy cost, using algorithms like Kadane's or K-Min Heap to ensure efficient operation.

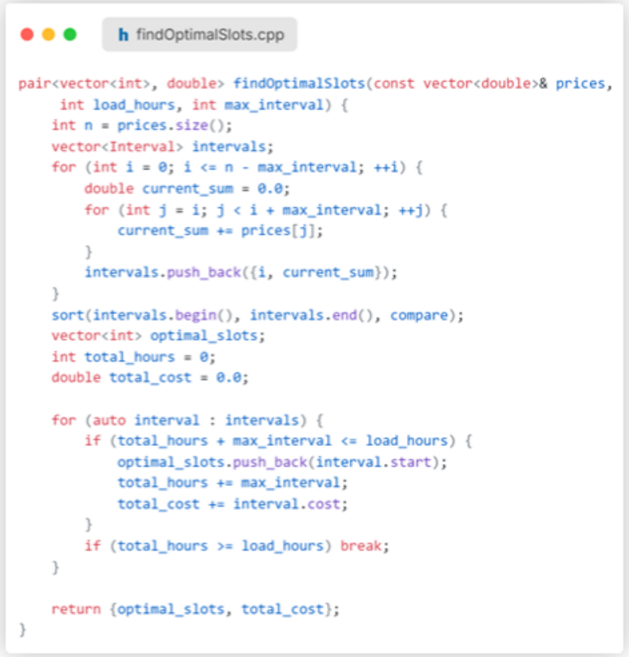
GitHub: https://github.com/ojhapiyush/HelioFlux

Energy Ninja - the Art of Smart Scheduling


Start
Device
Device.category == fixed
Prediction
Schedule == now
Device.category == Deferrable
findOptimalSlots (max_interval = load_hours)
Device.category == Regulatable
findOptimalSlots (max_interval = least_count)
Schedule
End



```cpp
vector<pair<int,int>> schedule(vector<double> &prices, int load_hours,
  int max_interval, double appliance_rating){

    pair<vector<int>, double> result = findOptimalSlots(prices, load_hours, max_interval);
    vector<int> optimal_slots = result.first;
    double total_cost = result.second;
    sort(optimal_slots.begin(), optimal_slots.end());

    vector<pair<int,int>> Scheduled_slots;
    for (int slot : optimal_slots) {
        Scheduled_slots.push_back({slot+1, slot+max_interval+1});
    }
    return Scheduled_slots;
}
```



```cpp
pair<vector<int>, double> findOptimalSlots(const vector<double>& prices,
     int load_hours, int max_interval) {
    int n = prices.size();
    vector<Interval> intervals;
    for (int i = 0; i <= n - max_interval; ++i) {
        double current_sum = 0.0;
        for (int j = i; j < i + max_interval; ++j) {
            current_sum += prices[j];
        }
        intervals.push_back({i, current_sum});
    }
    sort(intervals.begin(), intervals.end(), compare);
    vector<int> optimal_slots;
    int total_hours = 0;
    double total_cost = 0.0;

    for (auto interval : intervals) {
        if (total_hours + max_interval <= load_hours) {
            optimal_slots.push_back(interval.start);
            total_hours += max_interval;
            total_cost += interval.cost;
        }
        if (total_hours >= load_hours) break;
    }

    return {optimal_slots, total_cost};
}
```

Code Breakdown:

- **Optimal Scheduling Logic:**
  The findOptimalSlots() function identifies the cheapest time intervals by calculating cumulative costs over specified durations, sorting them using a minimum heap.
- **Scheduling Execution:**
  The schedule() function outputs optimal slots for running each appliance, adjusting based on its load category (e.g., Regulatable, Deferrable) and calculates the total energy cost.

GitHub: https://github.com/ojhapiyush/HelioFlux

Alert Mode - Your Early Warning System

Key Benefits

- Real-Time Alerts: Stay updated on tariff changes instantly.
- Cost Savings: Optimize energy usage and reduce bills.
- Multi-Channel Notifications: Receive important alerts on multiple platforms (App and WhatsApp).

System Features

- Automated Monitoring: Continuous tracking of tariff rates.
- Customizable Alerts: Users can choose notification preferences.
- Smart Decision Logic: Prioritizes notifications based on event severity.

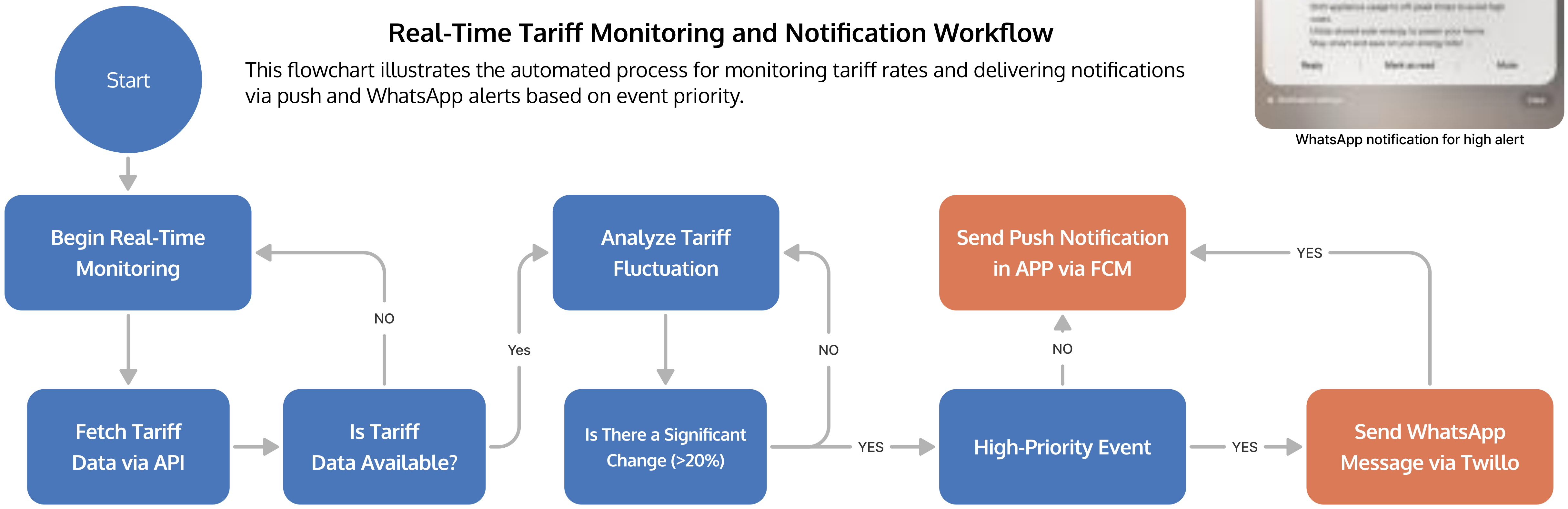
User Experience Highlights

- Ease of Use: Simple setup for notifications.
- Multi-Platform Reach: Alerts are accessible on mobile, web, and WhatsApp.
- Energy Optimization Tips: Recommendations included with notifications.

WhatsApp notification for high alert

Real-Time Tariff Monitoring and Notification Workflow

This flowchart illustrates the automated process for monitoring tariff rates and delivering notifications via push and WhatsApp alerts based on event priority.

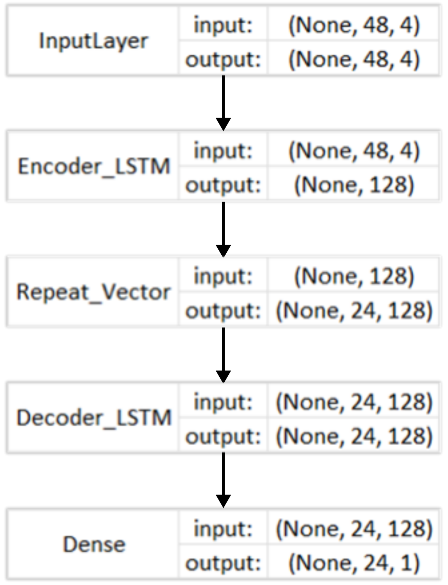


Start
Begin Real-Time Monitoring
Fetch Tariff Data via API
Is Tariff Data Available?
NO
Yes
Analyze Tariff Fluctuation
Is There a Significant Change (>20%)
NO
YES
High-Priority Event
NO
Send Push Notification in APP via FCM
YES
Send WhatsApp Message via Twillo
YES

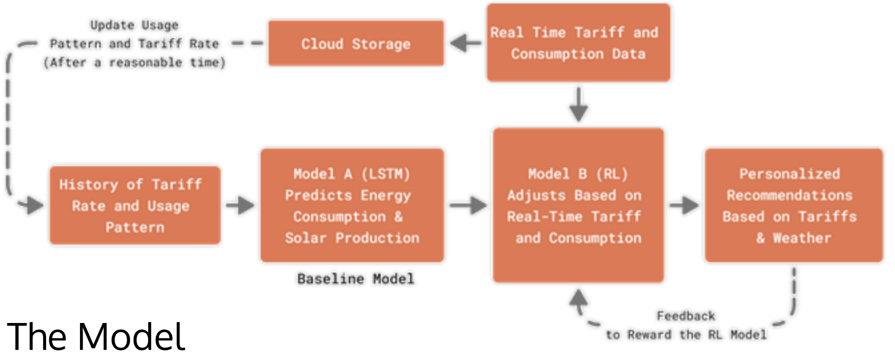
The Hybrid Brain - Making Hybrid Descision

This model uses LSTM as base model to predict energy consumption and solar production from historical data, with reinforcement learning (RL) refining predictions using real-time tariffs and consumption to generate personalized, tariff- and weather-optimized recommendations.

Objective: Predict tariff rates for the next 24 hours using historical data (weather conditions, grid load, and tariff rates) and act as baseline model.
Architecture: LSTM Encoder-Decoder model for time-series forecasting.

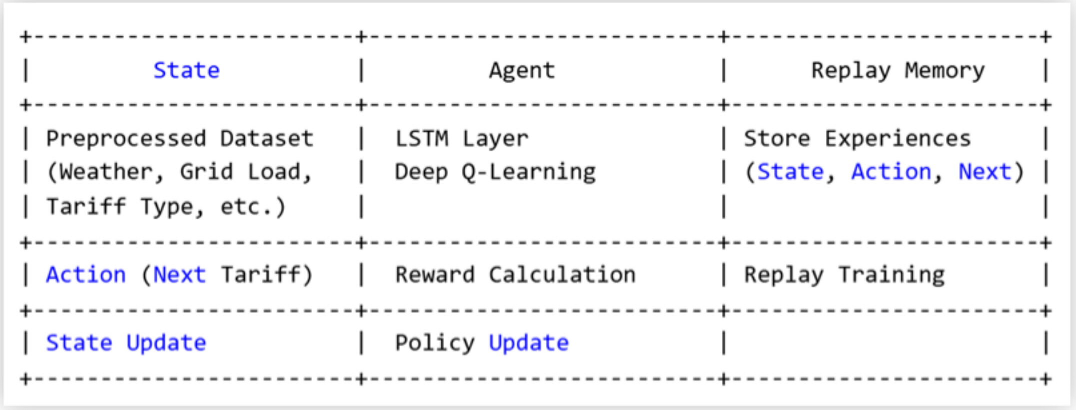


- **Input**: Past 48 hours.
- **Output**: Predicts tariff rates for the next 24 hours.
- Real-World Applications: Energy load management, tariff optimization based on weather conditions and energy demand.


Update Usage
Pattern and Tariff Rate
(After a reasonable time)
Cloud Storage
Real Time Tariff and Consumption Data
History of Tariff Rate and Usage Pattern
Model A (LSTM) Predicts Energy Consumption & Solar Production
Baseline Model
Model B (RL) Adjusts Based on Real-Time Tariff and Consumption
Personalized Recommendations Based on Tariffs & Weather
Feedback
to Reward the RL Model

The Model

Objective: Continuously predict and optimize electricity tariffs by learning from the past 48 hours of environmental and grid conditions, aiming to provide more precise and personalized cost management and tariff forecasting for each customer.
Architecture: A real-time Reinforcement Learning model using 48-hour data to predict the next 24 hours of electricity prices. The Deep Q-Learning agent refines predictions through state-action-reward updates, providing more personalized results than the baseline LSTM model.

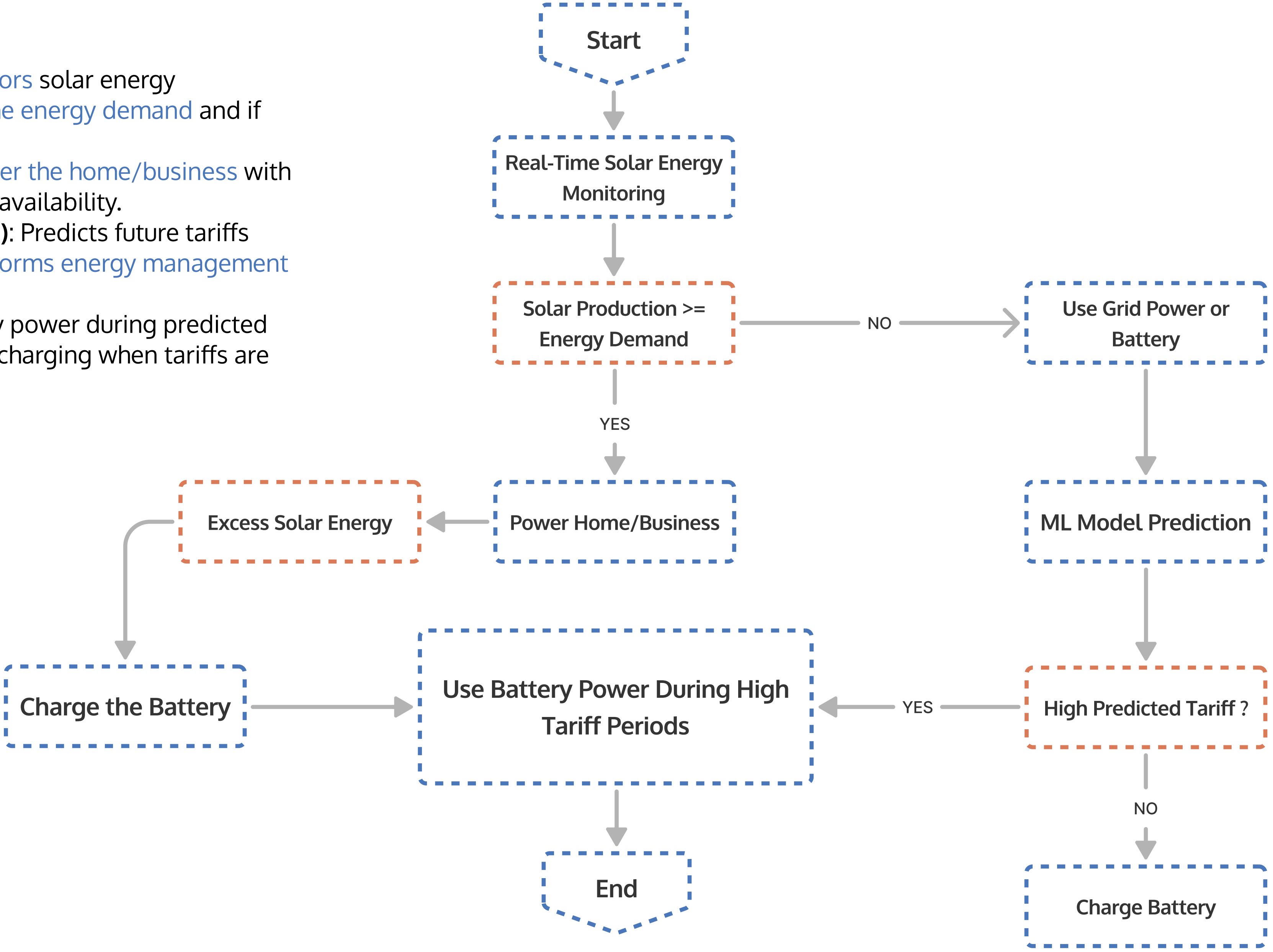

```
+-----------------------+--------------------------+----------------------+
|         State         |          Agent           |     Replay Memory    |
+-----------------------+--------------------------+----------------------+
| Preprocessed Dataset  | LSTM Layer               | Store Experiences    |
| (Weather, Grid Load,  | Deep Q-Learning          | (State, Action, Next) |
| Tariff Type, etc.)    |                          |                      |
+-----------------------+--------------------------+----------------------+
| Action (Next Tariff)  | Reward Calculation       | Replay Training      |
+-----------------------+--------------------------+----------------------+
| State Update          | Policy Update            |                      |
+-----------------------+--------------------------+----------------------+
```

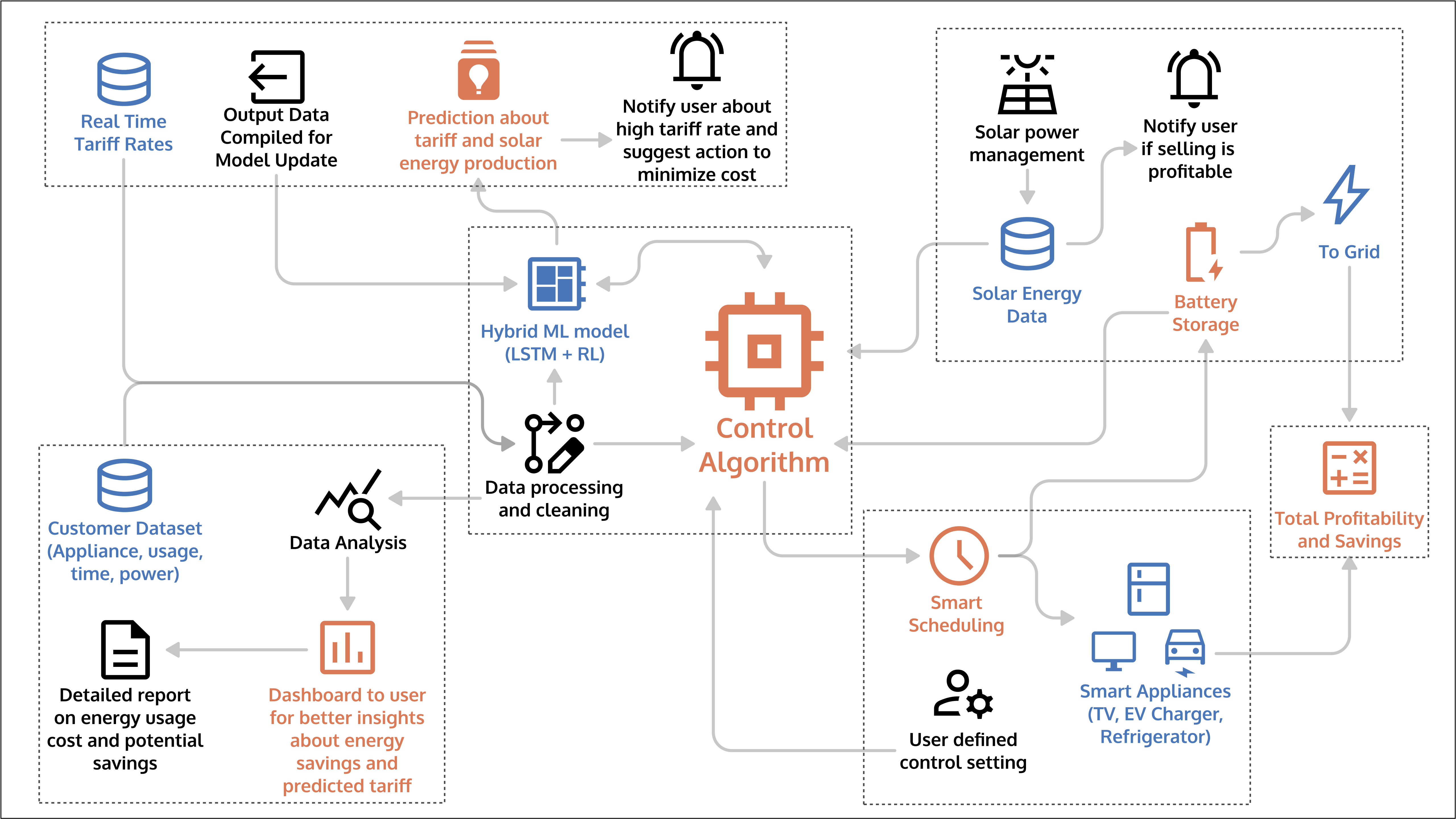
Solar Sentinel – Guarding Your Energy Source

- **Real-Time Solar Energy Monitoring**: Monitors solar energy production to determine whether it meets the energy demand and if there's excess.
- **Decision Logic**: Determines whether to power the home/business with solar or use the grid/battery based on solar availability.
- **Machine Learning Model (Tariff Prediction)**: Predicts future tariffs using historical and real-time data, which informs energy management decisions.
- **Battery Management System**: Uses battery power during predicted high tariff periods to optimize energy costs, charging when tariffs are lower or excess solar power is available.

Component	Description
Inputs	Resource Cost
	Resource Limit
	Demand Requirement
Objective	Minimize Total Cost
	Formula: resource_costs*resource_used
Constraints	Demand Constraint
	Resource Limits
Optimization Step	Use Linear programming
Results	Extract result for optimal allocation
	Display allocation for each resource

Resource Allocation Optimization Framework


Start
Real-Time Solar Energy Monitoring
Solar Production >= Energy Demand
NO
Use Grid Power or Battery
YES
Power Home/Business
Excess Solar Energy
ML Model Prediction
Charge the Battery
Use Battery Power During High Tariff Periods
YES
High Predicted Tariff ?
NO
End
Charge Battery


Real Time Tariff Rates
Output Data Compiled for Model Update
Prediction about tariff and solar energy production
Notify user about high tariff rate and suggest action to minimize cost
Solar power management
Notify user if selling is profitable
Solar Energy Data
Battery Storage
To Grid
Hybrid ML model (LSTM + RL)
Control Algorithm
Data processing and cleaning
Customer Dataset (Appliance, usage, time, power)
Data Analysis
Detailed report on energy usage cost and potential savings
Dashboard to user for better insights about energy savings and predicted tariff
Smart Scheduling
User defined control setting
Smart Appliances (TV, EV Charger, Refrigerator)
Total Profitability and Savings