DynamicPricingSystem

System Description

Abstract
This document provides the System Description (SysD) for the DynamicPricingSystem which determines optimal parking prices based on current occupancy, weather, time and event data. The system uses a reinforcement learning based model to maximize revenue while maintaining an optimal occupancy level.

Contents

1 Overview **3**
1.1 Significant Prior Art 4
1.2 How This System Is Meant to Be Used 4
1.3 System functionalities and properties 4
1.4 Important Delimitations 4

2 Services **5**
2.1 Produced Services 5
2.2 Consumed Services 5

3 Security **6**

4 References **6**

5 Revision History **7**
5.1 Amendments 7

1 Overview

This document describes the DynamicPricingSystem which calculates and updates parking fees dynamically using data from several other systems. The system continuously estimates demand based on occupancy, weather, time and event data and sets the price to maximize revenue. This is shown in figure 1.

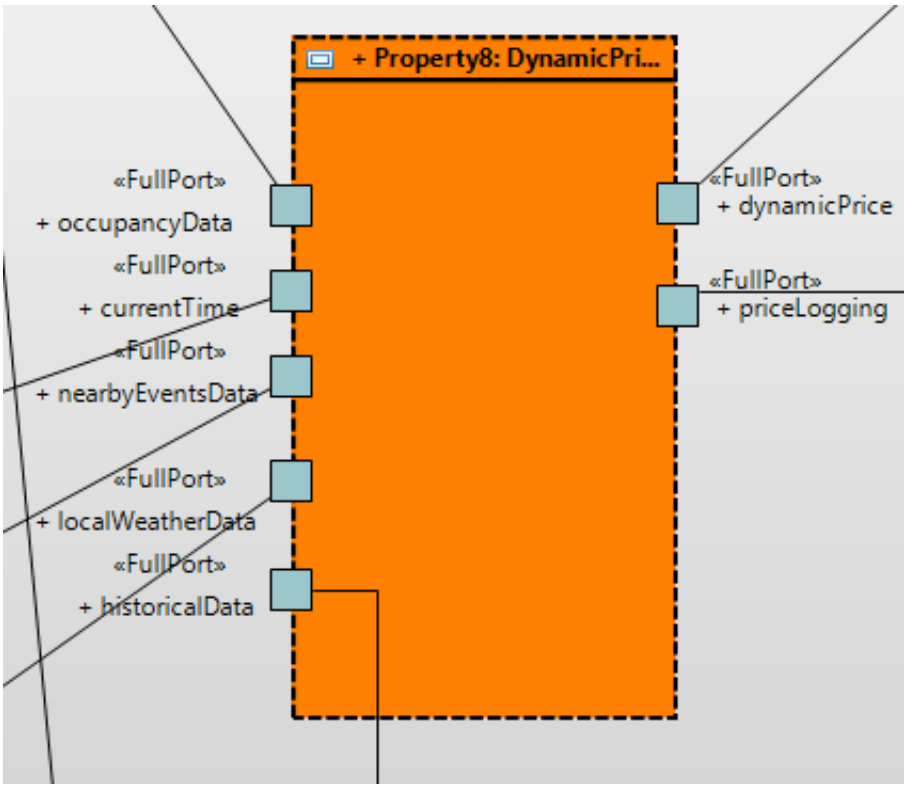

Figure 1: Use-case diagram showing the inputs and outputs of the DynamicPricingSystem.

The rest of this document is organized as follows. In Section 1.1 we reference major prior art capabilities of the system. In Section 1.2 we describe the intended usage of the system. In Section 1.3 we describe fundamental properties provided by the system. In Section 1.4 we describe delimitations of the system. In Section 2 we describe the abstract service functions consumed or produced by the system. In Section 3 we describe the security capabilities of the system.

1.1 Significant Prior Art

The DynamicPricingSystem builds on research in dynamic pricing and revenue management where AI and reinforcement learning have shown strong results in optimizing prices under changing demand. The system concept is similar to dynamic retail pricing approaches described in [1].

1.2 How This System Is Meant to Be Used

The DynamicPricingSystem is deployed inside the Arrowhead local cloud and operates as the central decision-making unit for parking prices. It receives continuous data streams from the OccupancySystem, WeatherSystem, TimeSystem and EventSystem, processes this information through its AI model and outputs updated parking prices. It also logs price changes and system states to the History Database for further training and evaluation.

1.3 System functionalities and properties

1.3.1 Functional properties of the system

The DynamicPricingSystem performs data aggregation, model training, model inference and decision making for price setting. It provides the DynamicPricing service to expose the current price and the PricingLog service to report decisions to the History Database.

1.3.2 Configuration of system properties

The system can be configured with different learning parameters, model types or pricing strategies. Configuration settings such as update frequency and target occupancy can be adjusted without changing system architecture.

1.3.3 Data stored by the system

The DynamicPricingSystem stores model weights, configuration files and recent data snapshots required for inference. Long term logs are sent to the History Database.

1.3.4 Non functional properties

- Security: Communication with other systems is encrypted and authenticated through HTTPS and TLS.
- Reliability: The system ensures consistent operation even when one or more data sources are temporarily unavailable.
- Latency: Pricing updates are calculated within minutes to an hour to reflect changes in occupancy and external conditions.
- Scalability: The system can handle additional input sources or model complexity as needed.

1.3.5 Stateful or stateless

The DynamicPricingSystem is stateful. It maintains its trained model, configuration state and recent operational data to ensure continuity between pricing cycles.

1.4 Important Delimitations

The DynamicPricingSystem does not perform payment processing or user interaction. It provides pricing information and logs but does not manage external communication beyond the Arrowhead local cloud.

2 Services

2.1 Produced Services

- DynamicPricing service
  Provides the current optimized parking price calculated from occupancy, weather, time and event data.
- PricingLog service
  Sends pricing decisions and model outputs to the History Database for storage and later training.
- updateConfig service
  Used to do changes in the configuration the dynamic pricing model, such as the targetted occupancy level or update frequency.

2.2 Consumed Services

- AggregatedOccupancy service
  Consumed from the OccupancySystem to monitor the number of parked vehicles.
- Weather service
  Consumed from the WeatherSystem to adjust pricing during weather changes.
- Time service
  Consumed from the TimeSystem to account for daily and weekly demand variations.
- EventSchedule service
  Consumed from the EventSystem to consider nearby events affecting demand.
- HistoricalData service
  Consumed from the History Database to retrain or fine tune the pricing model.

3 Security

Security is provided by the Arrowhead framework. All communication is handled through HTTPS with TLS encryption and Arrowhead-compliant authentication.

4 References

[1] M. Apte, K. Kale, P. Datar, and P. Deshmukh, “Dynamic retail pricing via q-learning – a reinforcement learning framework for enhanced revenue management,” 2024. [Online]. Available: https://arxiv.org/abs/2411.18261

5 Revision History

5.1 Amendments

No.	Date	Version	Subject of Amendments	Author
1	2025-10-14	1.0.0	Initial release	Mattias Öhman