ParkingLotPricingCloud

System Description

Abstract

This is the System of Systems Description (SoSD document), aimed at optimizing revenue of a parking lot through a dynamic pricing model.

Contents

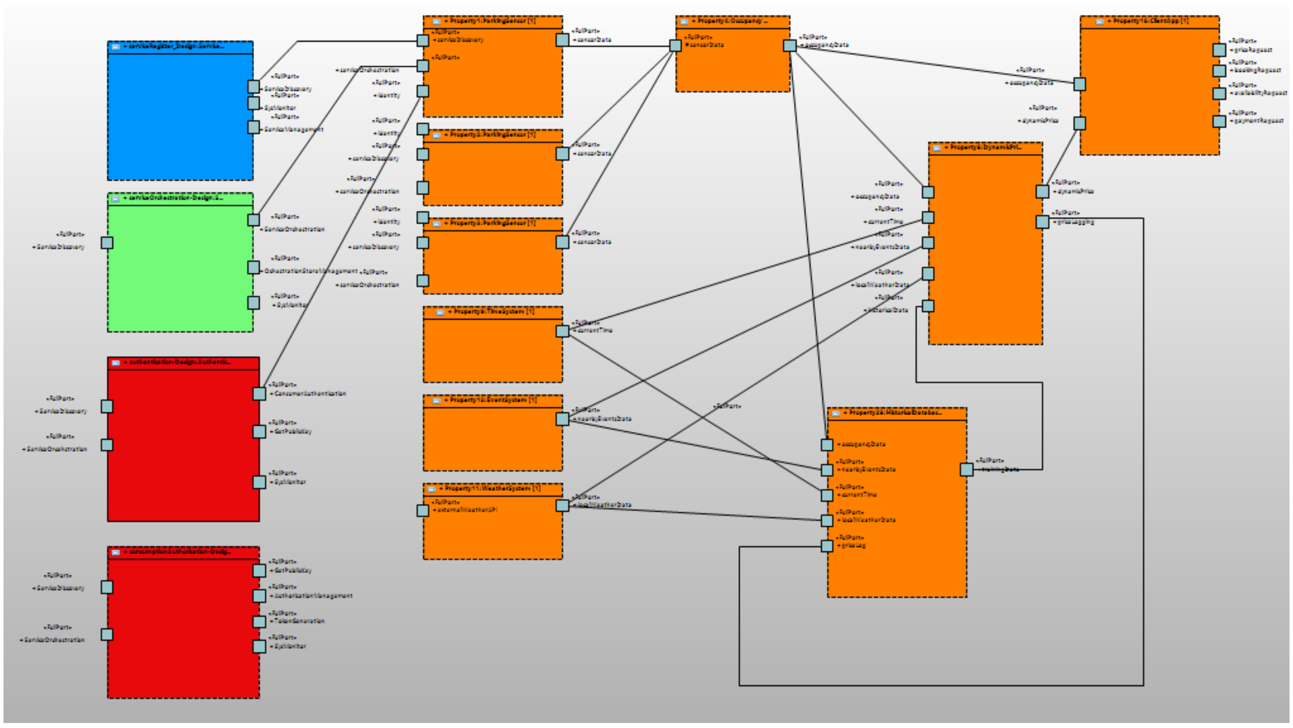

Figure 1: Overview of the ParkingLotPricingCloud

1 Overview

This document describes the parking lot pricing system of systems (SoS), which provides means for gathering and storing required data, training/storing/using AI/ML models to infer optimal prices, allowing customers to check availability, book and pay for parking spaces. This is illustrated by Figure 1, showing the whole cloud.

The rest of this document is organized as follows. In Section 1.1, we reference major prior art capabilitites of the SoS. In Section 1.2, we the intended usage of the SoS. In Section 1.3, we describe fundmental properties provided by the SoS. In Section 1.4, we describe de-limitations of capabilitites ofn the SoS. In Section 2, we describe the microsystem (abstract level with references to their SysDs) which constitutes the SoS. In Section 3, we describe the security capabilitites of the SoS.

1.1 Significant Prior Art

Although no specific system directly inspired this project, the rapid development of artificial intelligence and machine learning has enabled advanced methods for dynamic cost optimization. In particular, reinforcement learning approaches have demonstrated strong potential for adaptive pricing, surpassing traditional optimization techniques in terms of revenue generation [1].

1.2 How This SoS Is Meant to Be Used

The Dynamic Parking Pricing Lot (SoS) is intended to be deployed within a parking facility to optimize pricing and resource utilization in real time. In order to estimate demand, the SoS must continuously collect data from parking space sensors, weather services, event schedule and temporal information. Based on this data, the Dynamic Pricing System calculates and updates the parking fee dynamically, aiming to maximize revenue while maintaining an optimal occupancy level. End users interact with the system through a client application that provides real-time price information, availability and payment processing.

1.3 SoS functionalities and properties

1.3.1 Functional properties of the SoS

The SoS provides data-driven pricing for a parking facility by combining inputs from occupancy sensors, weather data, event schedules and temporal information. It continuously adjusts parking prices to balance occupancy and revenue, while enabling users to view availability and complete booking and payments through a client application.

1.3.2 Configuration of SoS properties

The SoS is configured to operate within a single Arrowhead Local Cloud, where each subsystem registers its services (e.g., sensing, pricing, payment) with the core systems. Service discovery and orchestration ensure dynamic connectivity and scalability as new sensors or data sources are added.

1.3.3 Data stored by the individual microsystem

Each microsystem stores only the data necessary for its function: The Dynamic Pricing System stores recent data needed during inference and also weights of its trained model. The History database maintains historical data required for training and evaluation.

1.3.4 Non functional properties

- **Security:** All communication is secured through the Arrowhead framework and HTTPS, ensuring that only authenticated systems can access or modify data.
- **Safety:** Fail-safe mechanisms prevent incorrect pricing or loss of service in case of subsystem failure.
- **Energy consumption:** Sensor units should be optimized for low power use and can operate efficiently in embedded environments. Training and AI inference consumes more energy.
- **Latency:** The system ensures low latency for client interactions and pricing responses, while sensor and environmental data are updated periodically (every 1–2 minutes), balancing responsiveness and efficiency.

1.3.5 Stateful or stateless

- Most services are stateless, processing each request independently (e.g., weather, time and event services).
- The Dynamic Pricing System and History database are stateful, preserving historical and learned data to improve long-term optimization and decision-making.

1.4 Important Delimitations

The SoS does not cover the payment system.

2 Services

2.1 Produced Services

From the perspective of the local cloud, the Dynamic Parking Pricing SoS produces user-facing services that external clients can access through the client application. These services allow interaction with the parking system and enable automated decision-making within the cloud.

- PriceRequest service
  Provides the current parking fee based on occupancy, time, weather, and event conditions.
- BookingRequest service
  Allows external users to reserve a parking spot when availability exists.
- AvailabilityRequest service
  Reports the current number of available spaces in real time.
- PaymentRequest service
  Enables users to initiate payments for parking sessions. The payment handling and confirmation are expected to be implemented by an external payment system.

2.2 Consumed Services

The SoS consumes several external data sources to support dynamic pricing and decision-making.

- ExternalWeatherAPI
  Provides environmental conditions such as temperature, rain, and snow level that influence parking demand.
- ExternalEventService
  Supplies information on local events that may affect parking demand.
- ExternalCalendarService
  Provides temporal data, including day of week and holiday information, used for modeling daily and seasonal demand variations.

All internal interactions between microsystems (e.g., sensor data aggregation, pricing, and logging) are handled within the local cloud and are not exposed externally.

3 Security

The Dynamic Parking Pricing SoS follows the Arrowhead Framework, which provides security through authentication, authorization, and encryption of all service interactions within the local cloud.

The local cloud exposes two main entry points. The first is the client application interface, which allows external users to access services such as price requests, bookings and availability information. These interactions are secured using HTTPS, and requests are authenticated and authorized through the Arrowhead Authorization System. The second entry point is the set of external data sources, such as weather, event and calendar APIs. These are accessed over standard HTTPS connections.

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		Mattias Öhman