Smart City System Architecture

Author: Eric Gieseke

Date: 9/15/2020

Introduction

This document provides the System Architecture for the Smart City Software System.

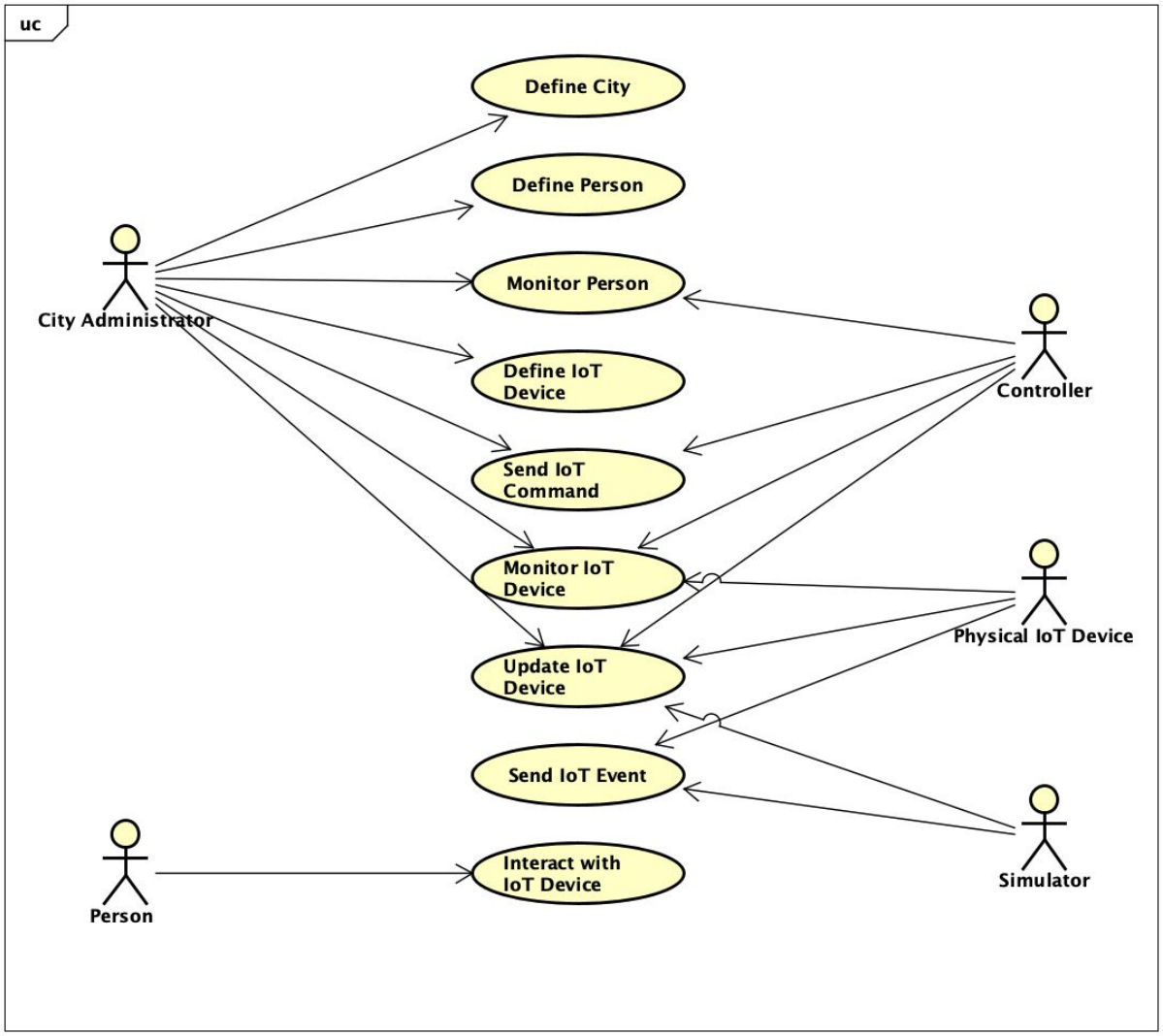
Overview

The Smart City Software System will fully automate the city of the future. Smart City allows city administrators to fully automate a city through AI-powered Internet of Things (IoT) devices, including cameras, microphones, robots, and other devices. Sensors monitor the location of persons within the city. Robot Assistants help residents and visitors, clean the city, and respond to emergencies. Some of the devices are controlled automatically, and all devices are able to interact with people through a voice interface.

Caption: Fully automated city enabled using the Smart City Software System.

Supported Use Cases

The following use case diagram documents the high-level use cases supported by the Smart City Software System.



Caption: UML Use case diagram with Smart City actors and use cases.

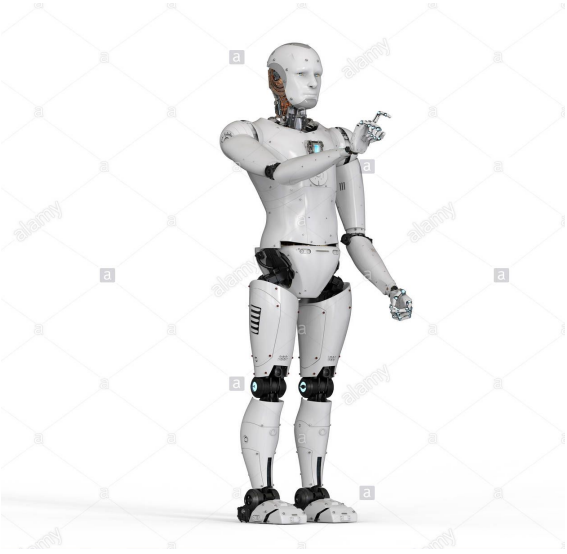
There are 5 types of actors:

- City Administrator
- Person
- Physical IoT Device
- Controller
- Simulator

City Administrators are responsible for configuring the smart city. This includes defining the city, provisioning the IoT devices, and setting up identities for the residents of the city.

Persons are the residents and guests that inhabit the city. Maximizing the person's experience is very important. Residents and visitors can interact with the various IoT devices and request services.

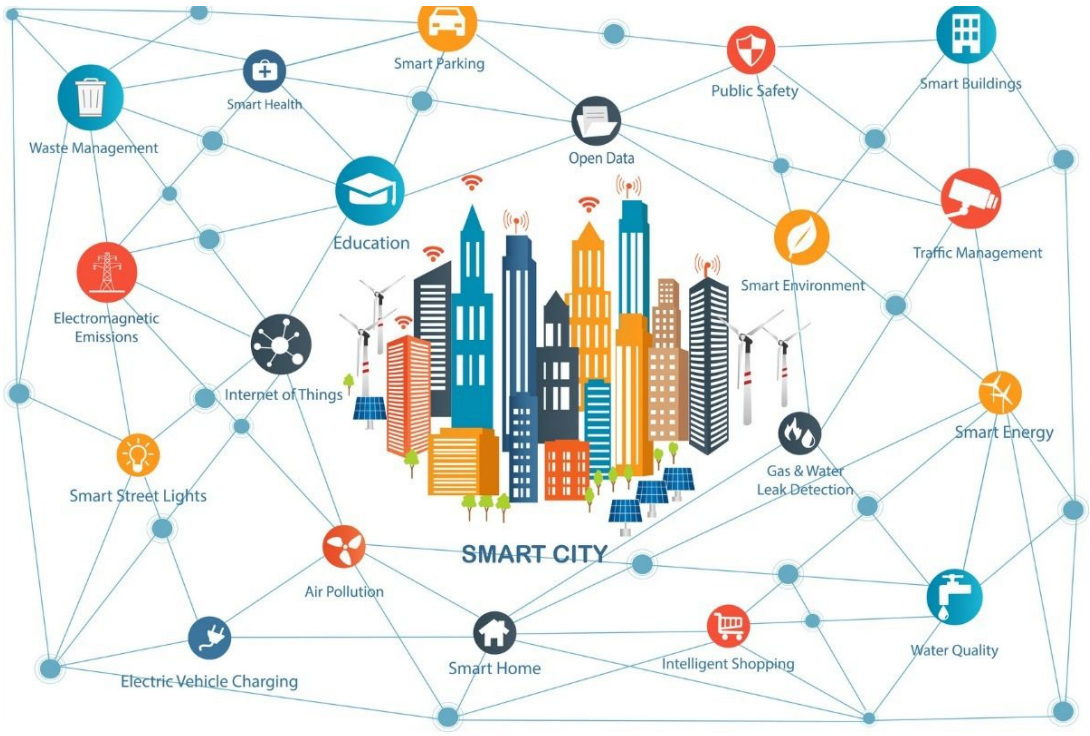
Physical IoT devices monitor the city. For example, one type of IoT device, the Robot Public Servant, maintains the city, including cleaning the city, assisting people, and responding to emergencies. IoT devices are fully automated and also respond to requests from persons and the Smart City Controller.



Caption: One type of IoT Device, the Public Servant Robot

The Smart City **Controller** provides the overall management for the city. The Controller monitors the city and people through IoT devices, processing events, and generating control commands.
The Controller monitors the location and status of all persons and manages the IoT devices around them either proactively or in response to voice commands. For example, when a person asks for help, the Controller will automatically send a robot public servant to assist the person.

The **Simulator** supports testing the Smart City system by providing a source for the sensor events in place of using actual physical IoT Devices.

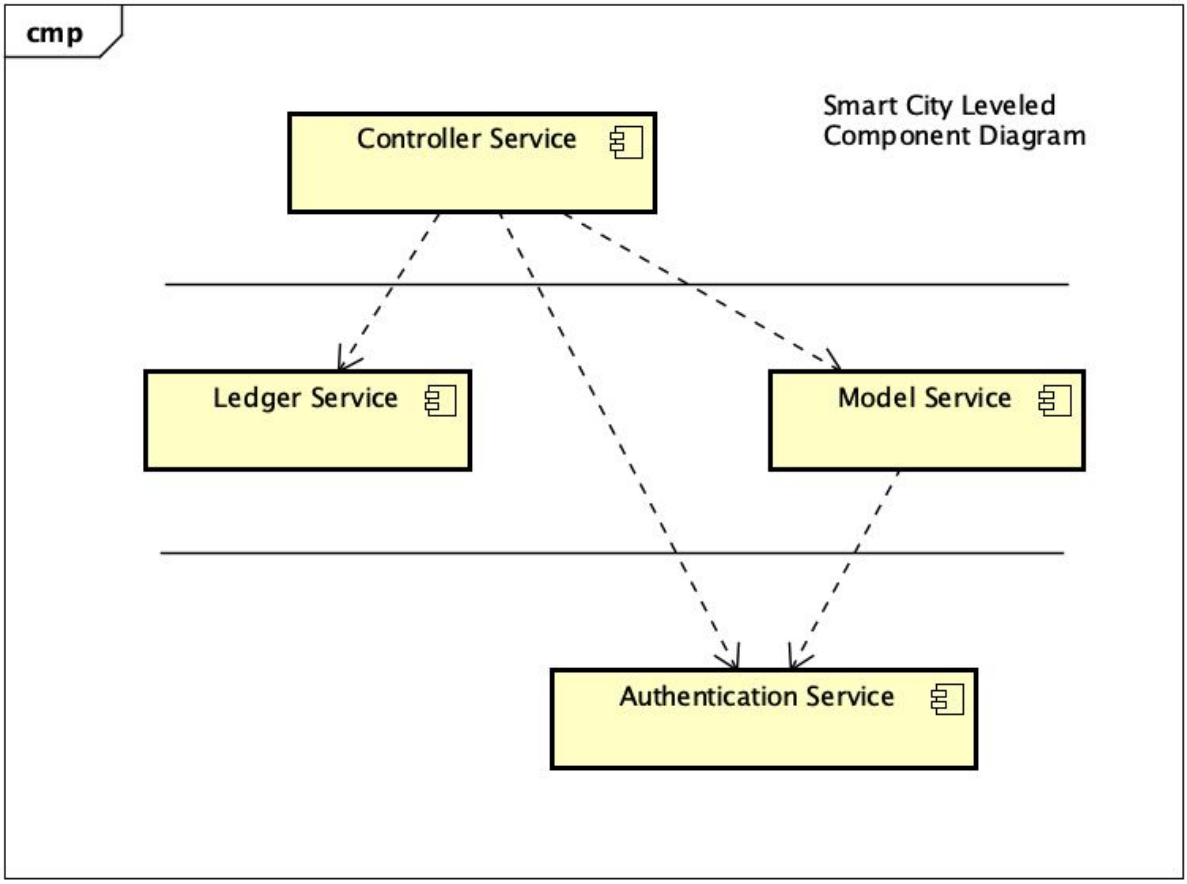


Caption: The Smart City System interconnects the city inhabitants with the Internet of Things.

System Architecture

This section defines the System Architecture for the Smart City Software System.

The following component diagram shows the high-level system components that make up the system.



Caption: Leveled UML Component diagram describing the high-level services for the Smart City Software System.

Smart City Model Service

The Smart City Model Service is responsible for managing the domain entities of the Smart City System. Domain entities include the city, people, and IoT devices, including street signs, information kiosks, street lights, parking spaces, public servant robots, and vehicles. The Smart City Model Service provides an API for interacting with those objects. The API supports querying the state of the entities, as well as updating the state.

The Model Service maintains the state of all domain objects.

Reference the Smart City Model Service Requirements and Design Document for design details (TBD in Assignment 2).

Smart City Controller Service

The Smart City Controller Service monitors the sensors and controls all of the IoT devices in the

city. The Controller Service is responsible for monitoring the events received from the IoT sensors located in the city and responding by appropriately controlling the IoT devices. In addition, the Controller Service is responsible for listening for voice commands and responding by controlling the appropriate IoT device. The Controller Service can also respond to general questions about the state of the city.

The Controller Service is responsible for maintaining the safety of the city and inhabitants. It is also capable of processing financial transactions to pay for services. It uses the Ledger Service (assignment 1) to manage transactions and account balances for persons, IoT devices, and the city.

Reference the Controller Service Requirements and Design Document for design details (TBD in Assignment 3).

Authentication Service

The Authentication Service manages the authentication of users and controls access to the devices in the city. The Authentication Service first identifies the user through biometrics (face and/or voice recognition). Once identified, the Authentication Service is used to gate access to control IoT devices.

Reference the Smart City Authentication Service Requirements and Design Document for design details (TBD in Assignment 4).

Technology

This section specifies the technology choices for the Smart City System.

The system will be implemented entirely in Java using the JDK 14.

The System Architecture follows a modular microservice architecture. Each of the 4 system components will be implemented as a microservice.

Each service will:

- Define a Java Interface that provides a list of all public operations supported by the Service
- Provide an implementation in Java of the service interface
- Provide a factory method for accessing a Singleton Instance of the service
- Fully encapsulate the service implementation details; anything external to the service may only access the public service interface

To simplify the implementation, each of the services will be collocated within the same Java

Virtual Machine (JVM). This will allow direct java level method access to the Service interfaces by the peer services and client applications.

References

https://en.wikipedia.org/wiki/Smart_city