Elevator Control System

Software Architecture Design

SAD Version 2.0

Team #3
8 October 2019

CS 460 Software Engineering

Contents

1 Introduction

Good software is identified by the end user for its features and functionality, by the client for its profitability and maintainability, and by the programmer for its legibility and clarity. It should be clear that all three pillars ultimately characterize good software, but more importantly that the three are interdependent. The developer must then guarantee all of the above for the sake of all entities involved. A top down approach has been taken up to this point. The feasibility study asked answered the fundamental question: Can it be done? The requirements definition document was then passed the baton and answered the next logical question: Within what parameters can it be done? The software specification document then answered: What is the desired behavior of the system? Now we may answer: How will ensure the desired behavior?

The ultimate goal is to ensure an efficient, safe, and maintainable implementation of the Cretaceous Gardens Controller (CGC) software[1]. To that end, all objects are illustrated in their proper contexts, and their crucial functions have been delineated as clearly as possible while simultaneously allowing the programmer enough flexibility so as to not stifle his or her creative process.

This is a road map for the eventual implementation of the system. It details the most relevant objects and their relationships in the form of diagrams. Explanations accompany all diagrams for the sake of clarity.

2 Design Overview

The class architecture presented here [2] *aims to maximize efficiency, maintainability, and safety. Without an efficient system, its safety may be compromised due to unnecessary delays between components. Maintainability can impact a safe implementation of the system if the system is permeable to programmer errors. The decoupling of concerns and a solid hierarchy are paramount. The design has been color-coded to increase readability. The colors are only for the sake of distinguishing one component from another.*

[1] Introduction by Zeke.

[2] Diagrams by Siri, Anas, Santi, and Zeke.

The component specification diagrams in the following section inherit these colors. Small red arrows point to objects that may be triggered by an event. Said objects are virtual devices and iconifications of their physical triggers have been connected to them with bidirectional blue arrows.

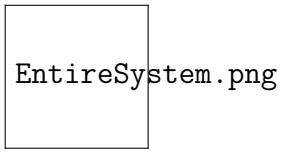

Figure 1: An overview of the Cretaceous Gardens Control System. Red arrows indicate that an object may be triggered by an event. The event is represented by the icon connected to the object.

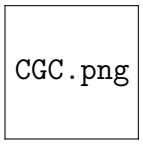

Figure 2: The Cretaceous Gardens Control System

3 Component Specifications

Here are the class specifications [3] *for objects found in Section 2. The most important attributes and functions are given and, where appropriate, any expected preconditions, parameters, return values, and post conditions are included. Explanations are given for each component, but it should be noted that helper functions are not shown, as they are entrusted with the programmer. Important fields and helper fields are treated analogously.*

[3] Component specifications by Anas and Siri

Some Class

Table 1: Some Caption

Attributes	
Elevators [0 .. 3]	Instances of virtual devices that represent elevators
Bay[1 .. 20	Instances of virtual devices that represent bays
currentMode	Indicates the mode of the system {alarmMode, emergencyMode, custodialMode, maintenanceMode, normalMode}

Functions

selectElevator()	→ selects best elevator upon request.
sendElevator(Elevator)	→ sends the elevator to the target bay.
checkSafetyControl(Bay, BayNumber)	→ performs a safety check on a bay.
checkSafetyControl(Elevator)	→ performs a safety check on a cabin.
changeMode()	→ toggles between normal mode and emergency mode.
openDoor()	→ opens cabin door and requests the same of bay analog
closeDoor()	→ closes cabin door and requests the same of bay analog
activateEmergencyMode()	→ explicitly enters emergency mode.
initialize()	→ initializes the system.
shutdown()	→ shuts down the system.
reset()	→ resets the system.

Figure 3: The Elevator Control System Class

Alarm Class

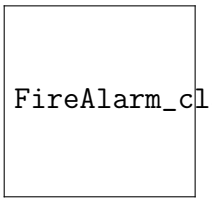

Figure 4: The Fire Alarm Class

Bay Class

The Bay class is a virtual device that mitigates request flow between the physical bay and the ECS.

Attributes

Message System	a data structure
bayNum	number of the bay
floorNum	number of the floor
bayButtons	instance of Bay Button Panel
Safety Control	instance of Safety Control
Bay Doors	Instance of Bay Doors

Functions

requestElevator(Direction)	→ requests an elevator specifying target floor direction.
buttonPressed()	→ listens for button presses.
openDoors()	→ opens bay doors.
closeDoors()	→ opens bay doors.

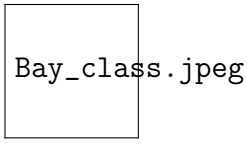

Figure 5: The Bay Class

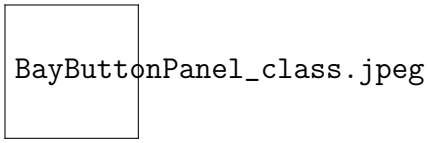

Figure 6: The Bay Button Panel Class

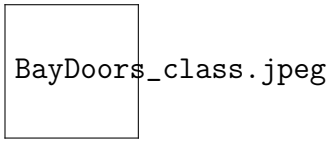

Figure 7: The Bay Doors Class

Elevator Class

The Elevator class encapsulates components that constitute the elevator, has communicates with physical components via virtual devices, and possesses some control logic for the cabin.

Attributes	
Cabin Number	an identifier
Current Floor	the floor on which the cabin resides
Direction	the direction in which the cabin is currently moving
Final Floor	the target floor
Cabin Buttons	an instance of Cabin Buttons class
Key Panel	an instance of Key Panel class
Safety Control	an instance of Safety Control class
Emergency Brake	a virtual device to represent the elevator brake
ElevatorMoving	indicates whether or not the cabin is in motion
ExecutiveMode	indicates whether or not an executive key is detected
SoundSystem	instance of Sound System class

Functions

validateEmergencyKey() → validates emergency key
validateExecutiveKey() → validates executive key
isJukeBoxPlaying() → indicates whether or not the jukebox is playing
setSpeakerConfig(Volume) → sets speaker volume
isIntercomActivated() → indicates whether or not intercom is active
changeIntercomStatus() → toggles intercom state
isFloorArrived() → indicates whether or not a request has been fulfilled
moveCabin(floor Number) → moves a cabin to the given floor
modifyDoors() → toggle door state
applyBrake() → activates the emergency brake
buttonPressed() → listens for a button press
emergencyCall() → sends emergency signal and starts the intercom

Figure 8: The Elevator Class

Figure 9: The Floor Display Class

Figure 10: The Emergency Brake Class

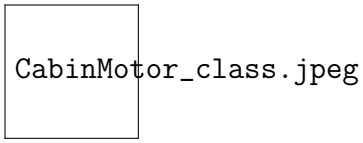

Figure 11: The Cabin Motor Class

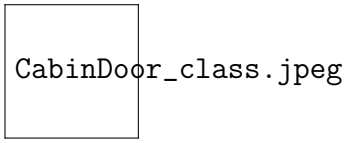

Figure 12: The Cabin Door Class

Cabin Sound System Class

The Cabin Sound System class is a virtual device that handles sound from the jukebox and intercom signals.

Attributes

Speaker	instance of virtual device that represents the speaker
Intercom	instance of virtual device that represents the intercom
JukeBox	instance of virtual device that represents the jukebox

Functions

isJukeBoxPlaying()	→ checks whether or not jukebox is playing
changeVolume(volume)	→ sets volume to the given value
isIntercomActivated()	→ checks whether or not the intercom is active
setIntercomStatus()	→ toggle intercom status
startPlaying()	→ play music

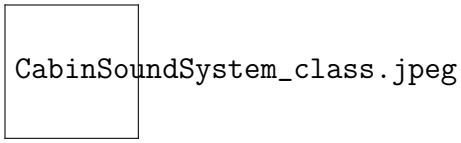

Figure 13: The Cabin Sound System Class

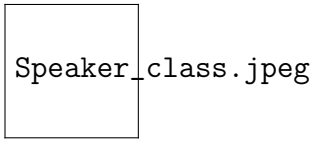

Figure 14: The Speaker Class

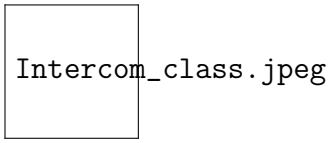

Figure 15: The Intercom Class

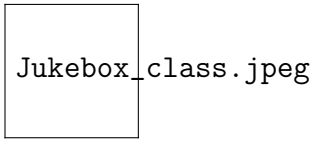

Figure 16: The Jukebox Class

Key Panel Class

The Key Panel class is virtual device to detect the presence of physical keys (emergency and executive).

Attributes

RFID Control	instance of virtual device that represents the RFID reader
Key Emergency Control	instance of Key Emergency class

Functions

isKeyDetected()	→ checks whether or not a key has been detected
getTypeKeyDetected()	→ returns the type of key detected

Figure 17: The Key Panel Class

Figure 18: The RFID Control Class

Figure 19: The Emergency Key Class

Cabin Button Panel Class

The Cabin Button Panel class is a virtual device to link the ECS with the physical button panel within the cabin.

Attributes

Alarm Button	instance of virtual device that represents the alarm button
Door Open Button	instance of virtual device that represents the door open button
Door Close Button	instance of virtual device that represents the door close button
Call Button	instance of virtual device that represents the call button
Floor Button	instance of virtual device that represents a floor button

Functions
getButtonPressed() → returns the button pressed

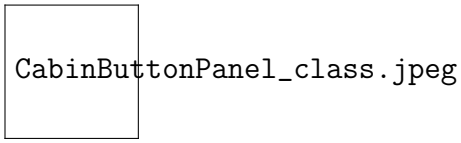

Figure 20: The Cabin Button Panel Class

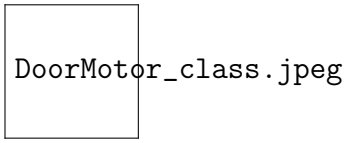

Figure 21: The Door Motor Class

Safety Control Class Class

The Safety Control class communicates with all safety hardware (motors, sensors, brakes, etc.) and holds the logic to meaningfully interpret signals from the hardware.

Attributes

Alignment Control	controls alignment sensors via virtual devices
Accelerometer Control	controls accelerometers via virtual devices
Weight Control	controls weight sensors via virtual devices
Speed Control	controls speedometers sensors via virtual devices
Light Curtain Control	controls light curtains via virtual devices
Alignment Control	

Functions
checkElevatorSafetyControls() → performs a system wide device check

Figure 22: The Safety Control Class

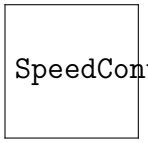

Figure 23: The Speed Sensor Class

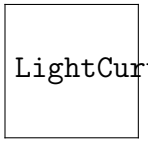

Figure 24: The Light Curtain Class

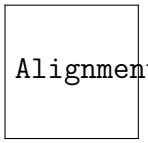

Figure 25: The Alignment Sensor Class

Figure 26: The Weight Sensor Class

Figure 27: The Weight Sensor Class

4 Sample Use Case

A broad overview of use cases begins this section and it is followed by detailed case descriptions. Human actors are denoted by small stick figures and have the same color scheme as the box that contains their labels. Guests of the hotel are colored green. An executive guest (denoted by a darker border than the one for the non-executive guest) subsumes the non-executive guest role.

ECSUseCaseUML.png

Figure 28: Shows the a general view of use cases as a diagram. The primary actors are color coded on the left, and exclusive goals share the same color scheme. Secondary actors are shown on the right. Goals that may be had by various actors have a grey color scheme. The dashed lines indicate inclusions.

Typical Scenarios

This section contains textual descriptions of the most common use cases for the ECS in detail. Travel from one floor to another is described for executive guests, non-executive guests. Guest communication with the front desk is also given, as are a few custodial scenarios.

Use Case: *GoToAnotherFloor*

Primary actor: Non-Executive Guest (NEG)

Goal in context: To travel from the current floor to another, non-executive, floor.

Preconditions: System is not in maintenance, nor emergency mode, nor alarm mode. Elevators, bays, and all mechanical parts can be in any safe state (aligned, below weight capacity, no obstructions, etc.).

Trigger: The non-executive guest decides to travel to another floor.

Scenario:

1. NEG observes the elevator bays.
2. NEG presses one or more buttons to request an elevator.
3. NEG enters first elevator to open its doors.
4. NEG presses one or more buttons to indicate the target floor (one or more button presses may have been erroneous)
5. NEG sees someone running toward the elevator
   (a) NEG presses the "open doors" button
   (b) NEG presses the "close doors" button
6. NEG travels to target floor with zero or more stops in between.
7. NEG exits the elevator upon arrival to the target floor.

Exceptions:

1. Alarm is triggered *before* entering elevator: NEG is denied access to all elevators.

2. Alarm is triggered *during* or *after* entering elevator: The "open door" button is disabled. ECS enters alarm mode. Emergency key must be inserted to put the ECS in emergency mode, from which normal mode may be accessed.

Priority: Essential, must be implemented

When available: Always

Frequency of use: Many times per day

Channel to actor: Via bay button panel, bay doors, elevator doors, cabin button panel

Secondary actors: Mechanical technician, ECS technician, Emergency Personnel, another generic human actor

Channels to secondary actors:

Mechanical Technician: cabin panel

ECS Technician: cabin panel

Emergency Personnel: alarm system, cabin panel, emergency key panel

Open Issues:

1. What happens if there is an obstruction while the ECS is in alarm mode?

Use Case: *GoToAnotherFloor*

Primary actor: Executive Guest (EG)

Goal in context: To travel from the current floor to the executive suite (top floor).

Preconditions: System is not in maintenance, nor emergency mode, nor alarm mode. Elevators, bays, and all mechanical parts can be in any safe state (aligned, below weight capacity, no obstructions, etc.).

Trigger: The EG decides to travel to his or her penthouse (the executive suite).

Scenario:

1. EG observes the elevator bays.
2. EG presses one or more buttons to request an elevator.
3. EG enters first elevator to open its doors.
4. EG inserts the executive key into the corresponding panel or scans the executive RFID card.
5. EG presses one or more buttons at least one of which is the newly activated penthouse button (one or more button presses may have been erroneous).
6. EG sees someone running toward the elevator
   (a) EG presses the "open doors" button
   (b) EG presses the "close doors" button
7. EG travels to target floor with zero or more stops in between.
8. EG exits the elevator upon arrival to the target floor.

Exceptions:

1. Alarm is triggered *before* entering elevator: EG is denied access to all elevators.
2. Alarm is triggered *during* or *after* entering elevator: The "open door" button is disabled. ECS enters alarm mode. Emergency key must be inserted to put the ECS in emergency mode, from which normal mode may be accessed.
3. invalid executive key or executive RFID card: EG must contact hotel staff to resolve the issue.

Priority: Essential, must be implemented

When available: Always

Frequency of use: Many times per day, but fewer than the analogous NEG scenario

Channel to actor: Via bay button panel, bay doors, elevator doors, cabin button panel, key panel

Secondary actors: Mechanical technician, ECS technician, Emergency Personnel, another generic human actor

Channels to secondary actors:

Mechanical Technician: cabin panel

ECS Technician: cabin panel

Emergency Personnel: alarm system, cabin panel, emergency key panel

Generic human actor: bay doors, cabin doors

Open Issues:

1. How to deal with NEG that may "piggyback" on the EG's access?

Use Case: *CallFrontDesk*

Primary actor: Non-Executive Guest (NEG)

Goal in context: To communicate with the front desk personnel.

Preconditions: Intercom system is functioning properly.

Trigger: The NEG has a question, comment or concern while inside an elevator cabin.

Scenario:

1. NEG observes "call button."
2. NEG presses the button and waits for someone to answer.
3. NEG has a conversation that addresses the question, comment, or concern.
4. NEG ends the conversation.
5. NEG continues traveling in the cabin or exits at some floor.

Exceptions:

1. Front Desk Personnel does not answer: intercom call times out. NEG may try again.
2. Alarm is triggered: alarm mode is entered, intercom remains functional.

Priority: Essential, must be implemented

When available: Always

Frequency of use: A moderate number of times per day

Channel to actor: Via bay button panel, intercom system

Secondary actors: Front desk personnel

Channels to secondary actors:

Front Desk Personnel: intercom system

Open Issues:

1. How are calls from multiple cabins handled?

Use Case: *CleanCabin*

Primary actor: Custodial Personnel (CP)

Goal in context: To perform cabin cleanliness upkeep or to clean spills.

Preconditions: System is not in maintenance, nor emergency mode, nor alarm mode. Elevators, bays, and all mechanical parts are in a safe state (aligned, below weight capacity, no obstructions, etc.).

Trigger: CP is scheduled to perform upkeep or a spill occurs within a cabin.

Scenario:

1. CP is notified of the cabin in question.
2. CP presses button to request the specific elevator.
3. CP enters the cabin and waits for non-custodial personnel to exit the cabin.
4. CP inserts custodial key to enter custodial mode.
5. CP performs routine cleaning or spill.
6. CP removes custodial key.
7. CP exits a spotless cabin.

Exceptions:

1. Maintenance key is invalid: custodial personnel must acquire a valid custodial key.
2. Alarm is triggered: alarm mode is entered and custodial key functions are overridden.

Priority: Essential, must be implemented

When available: Always

Frequency of use: On regularly scheduled intervals and on random occasion

Channel to actor: Via bay button panel, bay doors, elevator doors, key panel, cabin button panel

Secondary actors: Mechanical technician, ECS technician, Emergency Personnel, another generic human actor.

Channels to secondary actors:

Mechanical Technician: cabin panel

ECS Technician: cabin panel

Emergency Personnel: alarm system, cabin panel, emergency key panel

Generic Human Actor: bay doors, cabin doors

Open Issues:

1. None known.

Use Case: *MoveGuestWasteAndLaundry*

Primary actor: Custodial Personnel (CP)

Goal in context: To transfer guest laundry and guest waste while minimizing visibility.

Preconditions: System is not in maintenance, nor emergency mode, nor alarm mode. Elevators, bays, and all mechanical parts are in a safe state (aligned, below weight capacity, no obstructions, etc.).

Trigger: Custodial waste basket is full and/or custodial laundry basket is full.

Scenario:

1. CP runs out of space for guest waste and laundry in custodial cart.
2. CP observes elevator bays.
3. CP requests elevator and waits for one without guests.
4. CP inserts custodial key to enter custodial mode.
5. CP travels directly to the floor where the laundromat and waste disposal area are located.
6. CP removes custodial key.
7. CP exits cabin.

Exceptions:

1. Maintenance key is invalid: custodial personnel must acquire a valid custodial key.
2. Alarm is triggered: alarm mode is entered and custodial key functions are overridden.

Priority: Optional, may be implemented

When available: Always

Frequency of use: Varies with guest volume. The more guests, the more often this is used.

Channel to actor: Via bay button panel, bay doors, elevator doors, key panel, cabin button panel

Secondary actors: Mechanical technician, ECS technician, Emergency Personnel, another generic human actor

Channels to secondary actors:

Mechanical Technician: cabin panel

ECS Technician: cabin panel

Emergency Personnel: alarm system, cabin panel, key panel

Generic Human Actor: bay doors, cabin doors

Open Issues:

1. Should custodial personnel use separate elevators for this?

Rare Scenarios

This section features scenarios that may occur occasionally. This includes mechanical and technical repair or maintenance and emergency situations.

Use Case: *RepairElevator*

Primary actor: Mechanical Technician (MT)

Goal in context: To fix mechanical problems along the shaft, within the bay, within the cabin, or among any motors.

Preconditions: System is not in maintenance, nor emergency mode, nor alarm mode. Elevators, bays. Mechanical state may or may not be in a safe state.

Trigger: A mechanical failure occurs.

Scenario:

1. MT is informed of the location of the problem.
2. MT uses stairs to access entry point to the system.
3. MT inserts maintenance key.
4. MT fixes the issue.
5. MT calibrates and resets sensors if necessary.
6. MT tests the elevator by moving it through various floors.
7. MT removes maintenance key and restores it to its normal functioning mode.
8. MT exits the shaft, bay, or cabin where issue was found.

Exceptions:

1. Alarm is triggered *before* entering elevator: MT is denied access to all elevators.
2. Alarm is triggered *during* or *after* entering elevator: The "open door" button is disabled. ECS enters alarm mode. Emergency key must be inserted to put the ECS in emergency mode, from which normal mode may be accessed. MT inserts maintenance key so emergency personnel will be aware of the situation.
3. Maintenance key is invalid: Mechanical Technician must resolve the issue with hotel staff.

Priority: Essential, must be implemented

When available: Always

Frequency of use: A few times per month

Channel to actor: Via bay button panel, bay doors, elevator doors, cabin button panel, key panel

Secondary actors: Mechanical technician, Emergency Personnel, another generic human actor

Channels to secondary actors:

Mechanical Technician: cabin panel

Emergency Personnel: alarm system, cabin panel, emergency key panel

Generic human actor: bay doors, cabin doors

Open Issues:

1. How to handle alarm mode that is triggered during mechanical maintenance?

Use Case: *RepairOrConfigureECS*

Primary actor: ECS Technician (ET)

Goal in context: To fix any issues related to the ECS or to configure it.

Preconditions: System is not in maintenance, nor emergency mode, nor alarm mode. Elevators, bays. Mechanical state is presumed to safe. ECS states may or may not be in a safe state.

Trigger: An ECS failure occurs or configuration is sought.

Scenario:

1. ET is informed of problem or prompted to configure the ECS.
2. ET accesses the ECS which is centralized somewhere in the building.
3. ET uses intercom to clear all cabins before disabling them and all bays.
4. ET fixes the issue or configures the system.
5. ET runs tests and performs any necessary calibration.
6. ET starts the system and leaves the hotel.

Exceptions:

1. Alarm is triggered *before* disabling the system: ET must evacuate with everyone else.

2. Alarm is triggered *during* or *after* disabling the system: ET must abort the operation and evacuate.
3. Access denied: ET must resolve the issue with the software company.

Priority: Essential, must be implemented

When available: Always

Frequency of use: A few times per month

Channel to actor: Via direct access to the ECS

Secondary actors: Generic human actor

Channels to secondary actors:

Emergency Personnel: alarm system

Generic human actor: intercom

Open Issues:

1. How to handle mechanical failure including ECS failure?

Use Case: *SearchAndRescue*

Primary actor: Emergency Personnel (EP)

Goal in context: Locate endangered human beings within the building as soon as possible.

Preconditions: System is alarm mode. Mechanical and ECS states may or may not be safe.

Trigger: The alarm system is triggered.

Scenario:

1. EP receives alarm signal.
2. EP rushes to to the building.

3. EP accesses first available elevator, if any.
4. EP inserts emergency key.
5. EP controls the elevator to minimize search and rescue time.
6. EP resolves endangerments after some indefinite repetition of the previous step.
7. EP removes the emergency key and restores the system to its normal functioning state.

Exceptions:

1. Access denied: EP must use the stairs.

Priority: Essential, must be implemented

When available: Always

Frequency of use: A few times per year

Channel to actor: Via cabin key panel

Secondary actors: Generic human actor

Channels to secondary actors:

Generic human actor: bay doors, cabin doors

Open Issues:

1. How to handle mechanical failure or ECS failure during emergency operation?

5 Design Constraints

Due to the real-time nature of the system, there exist some additional constraints[4]*. Namely, it must be the case that all data structures concerning the safety controls are as fast as possible but also that they are capable of prioritizing all signals in the best way possible.*

[4]Design Constraints by Anas.

5.1 Safety

The safety is highly prioritized in our design of the ECS. We have considered associating the fire alarm directly with the ECS because in the case of the fire alarm triggering, this event has a very high priority and the ECS should immediately react to this event. When it comes to elevator and bay safety, each elevator and bay will have set of Safety Controls that they can communicate through and the ECS monitors the situation from the top. By ensuring safety for both bays and the elevators, the ECS operation will be carried smoothly without hurting the passengers.

5.2 Implementation Guidelines

According to the design, we suggest programmers to use some sort of Concurrent safe Messaging Queue for the communication between sensing objects and their associated parent objects. We also recommend using concurrent safe Priority Queue for the ECS, so the ECS can react based on the certain given priority. The priority should be considered because in the case of an emergency, the priority for that event should be at the very top so that the ECS should immediately react to it by closing its doors and going to first floor if its not there already.

6 Definition of Terms

The following is a list of definitions contain the most commonly used technical terms within this document, whose meaning may not be immediately apparent to the lay reader. Most definitions are defined by the authors for use within the context of this document. Some may originate from vocabulary shared across the general references cited . In the event that a definition was taken directly from a source, it is followed by a citation. [5]

CGC: Acronym for Cretaceous Gardens Controller

DVR: Acronym for Digital Video Recorder

Electrical Conduction: The movement of electrically charged particles through a transmission medium.

[5] This list is mostly a reduction of the term list found in the preceding Software Design Specification document.

GPS: Global Positioning System

Hardwired Ethernet: This references the latest IEEE standard for Ethernet utilizing physical cables.

Network: All nodes with which the CGC interacts, the links that connect them to each other and to the CGC, the CGC itself, and all related databases.

Node: The generic term that refers to any device connected to the CGC in any way. This includes autonomous vehicles, tokens, the T.Rex monitor, all electric fence panels, all kiosks, and all cameras.

Safely Inactive: A state in which a vehicle is fully functional and ready to be dispatched.

Safely Occupied: A state in which a vehicle contains at least one person, is locked, and is ready to depart.

Token: An interactive device used by the visitor that grants access to locations.