Bakers Rack High Level Design

Primary Owner ots-dse-hw-team (LDAP)

Last modified 4 months ago by mrshlr.

Bakers Rack High Level Design

Overview / Problem Statement

Current State

We need a quick and scrappy solution to prove out REK by creating a system which enables the return of packages via barcode scan. The time, date, WHID, and tracking number of said package must be collected and stored in a DB. Currently we do not have this.

Problem To Solve

1. Proving the development process and value of quick package drop off locations

2. Creating an incentive to grow for project REK

Requirements

- Must capture the following metrics upon package scan
  - WHID
  - Tracking number
  - Time
  - Date
- Must capture capacity metrics
- Must provide a secure location for the package returned

Assumptions

- We will be leveraging existing firmware and PCB boards used in Deadbolt
- We will be leveraging ELO Device and CMS system
- There will be internet access available for the ELO device

Proposed Solution

Given the time constraints, the proposed solution will come with tech debt. That being said, we will be leveraging the existing Deadbolt Firmware and PCB boards we can control the opening and closing of the door solenoid and measure the relative capacity of a given Bakers Rack using the PCB sensors as capacity gauges. One PCB board would wire up 5 separate sensors (as they do for the slot) but instead of measuring whether a device is there, they would be spaced out evenly in fifths (view image below) and when a sensor is triggered, the capacity for the Bakers Rack would be updated accordingly. The frontend UI would be Android, leveraging the existing Deadbolt Application but stripped of the majority of the core functionality needed in the Deadbolt Smart lockers, using only what is needed for Bakers Rack. The backend will be a simple Dynamo DB table with AppSync API to capture the bare details listed above. View the Architecture below for details.

Hardware Visual
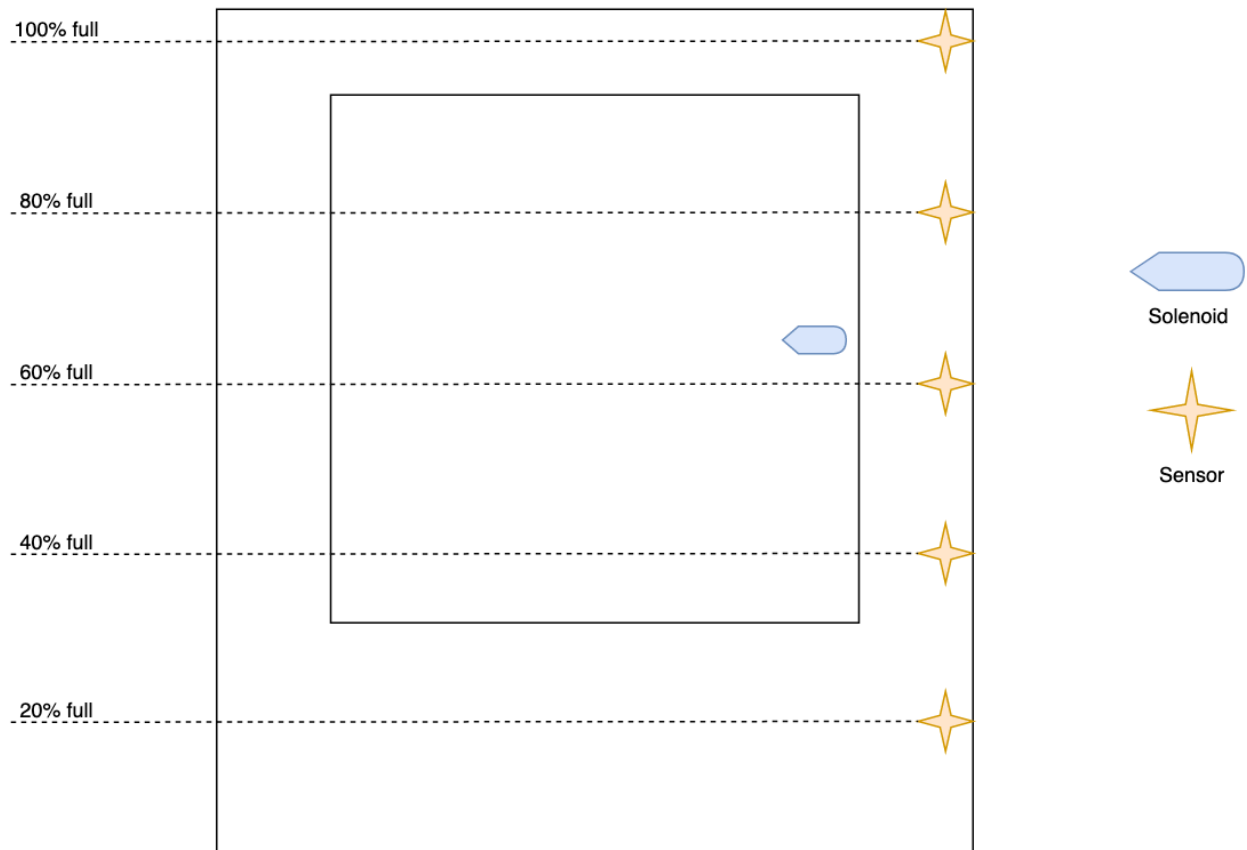
Architecture

Edit Source Diagram | Full Screen

Baker Rack Backend
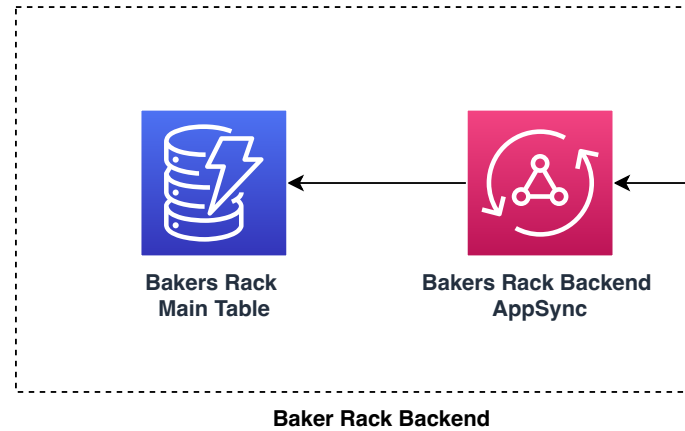
MQTT Topic Mapping

Topic	Raw Message Returned	Raw Message Published	How to handle
"shelf-v1/<MAC>/shelf-door-release"	"hold"	"release"	Opens the door with the given <MAC>
"shelf-v1/<MAC>/info"	"shelf-door:<position> cabinet-door:<position> fault:<csv list of error codes>"	N/A	Parse error codes, if any are fatal or bad throw exception
"shelf-v1/<MAC>/slot<N>/info"	"status:<status> current:<current> time: <time on charge> charge:<charge delivered>"	N/A	While other metrics are returned we only care about the "proximity" value. If it returns "present" we know that the Bakers Rack is filled up to that sensor
"controller-v1/status"	"online" | "(offline)"	N/A	Throw an exception (means that the PCB is offline)

APIs

Slot queries are a simple query on GSI 1 with a given locker ID. No sequence diagram provided.

AddPackageScan

Request

```
{
  "locker_id": "abc",
  "whid": "xyz",
  "__metadata": {
    "timestamp": 1234567890,
  }
}
```

Response

```
{
  "message": "successfully logged package scan",
  "success": true,
}
```

Data Storage

Supported Access Patterns

Access Pattern	Query Type	Required Parameters
Get all package scans by WHID (with time box)	1 Base query	WHID
Get time/date of a tracking scan	1 Base query	WHID, tracking

Baker Rack Main Table

Primary key		Attributes	
Partition key: pk	Sort key: sk		
SCAN#usf2	TIMESTAMP#56787656	data	
		TRACKING#234543456	
	TIMESTAMP#98765678	data	
		TRACKING#123456789	
SCAN#ewr3	TIMESTAMP#76545677	data	
		TRACKING#987678987	

Tags: