Datto Disaster Tracking

Design/Architecture Document

Masters of Disaster
Nsama Chipalo, Brandon Cole, Aaron Damrau, Jhossue Jimenez, Jacob Peterson

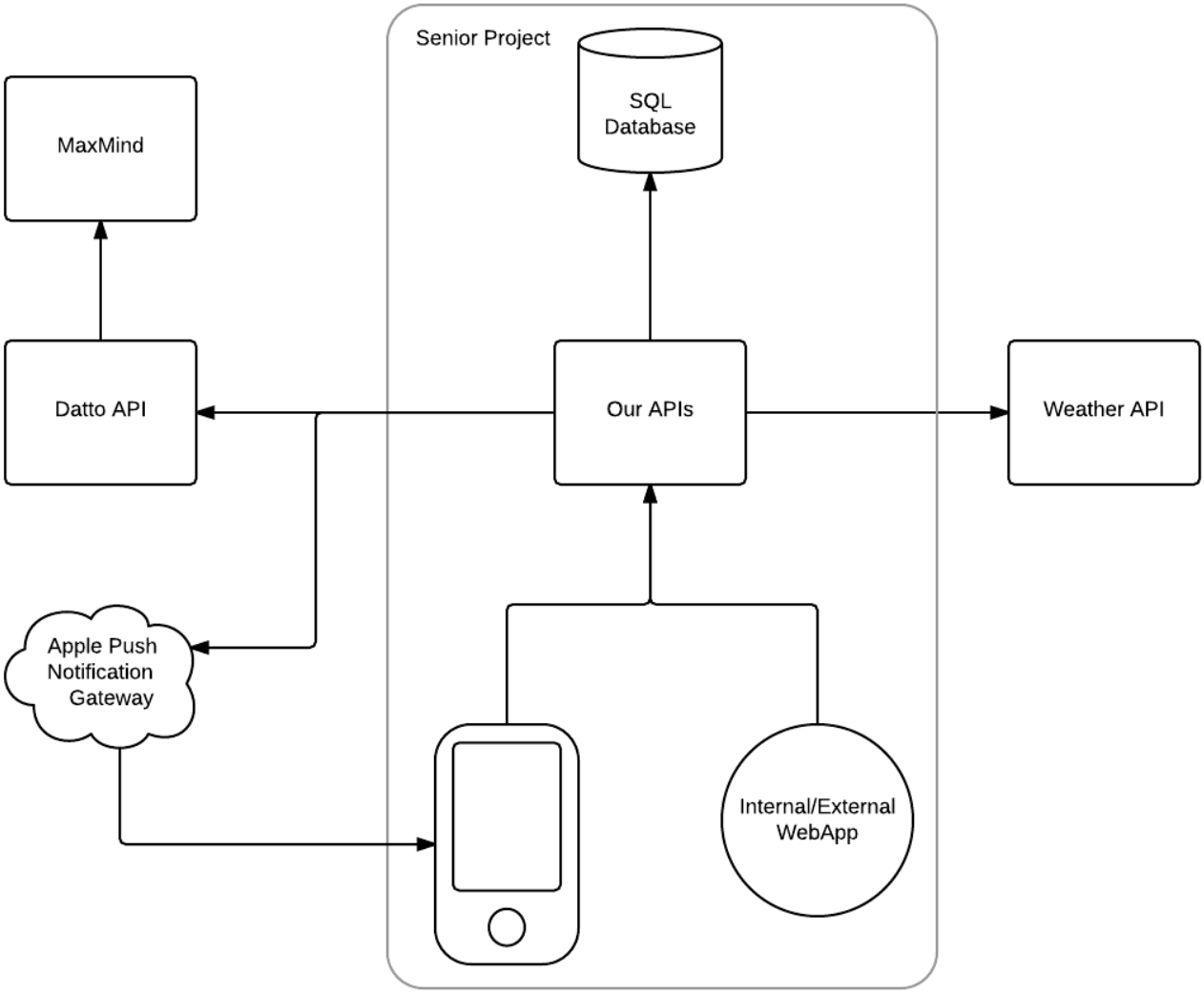
Last Updated
October 6th, 2014

Table of Contents

Purpose

This document provides a comprehensive architectural overview of the system. It is intended to capture and convey the significant architectural decisions which have been made on the system.

High Level Architecture Diagram



External:

- MaxMind - provides geolocation information based on IP addresses to Datto.
- Datto API - provides APIs for our project to get a list of customers and provides login functionality.
- Weather API - provides weather information to be used for risk calculation.
- Apple Push Notification Gateway - responsible for sending push notifications to the corresponding MSP owned Apple devices.

Internal:

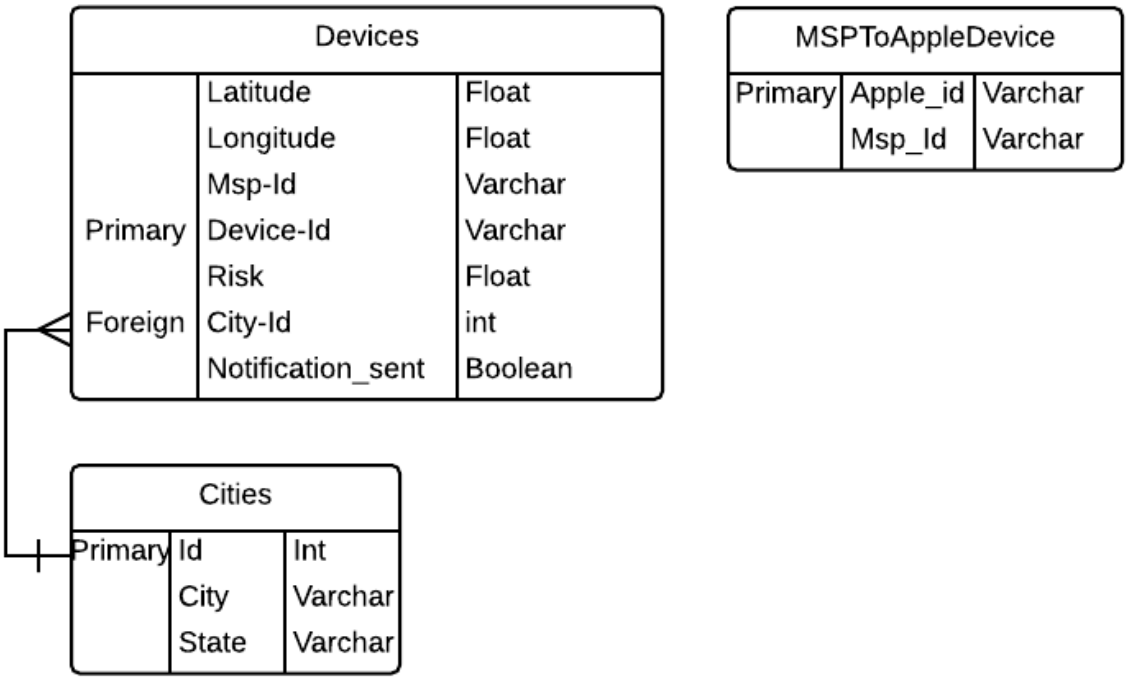
- Mobile Device - used by MSPs, shows a list of their customers. Notifies the MSP when one or more of their devices are above a risk threshold.
- Internal/External Web App - Used by Datto and MSPs, renders a map view of devices.

- Our APIs - Used by the Web App and Mobile Devices to render views and lists. Responsible for all logic that occurs (calculating risk levels) and coordinating communications with external APIs.
- SQL Database - Used as a cache that is updated periodically with new weather and device information.

Database

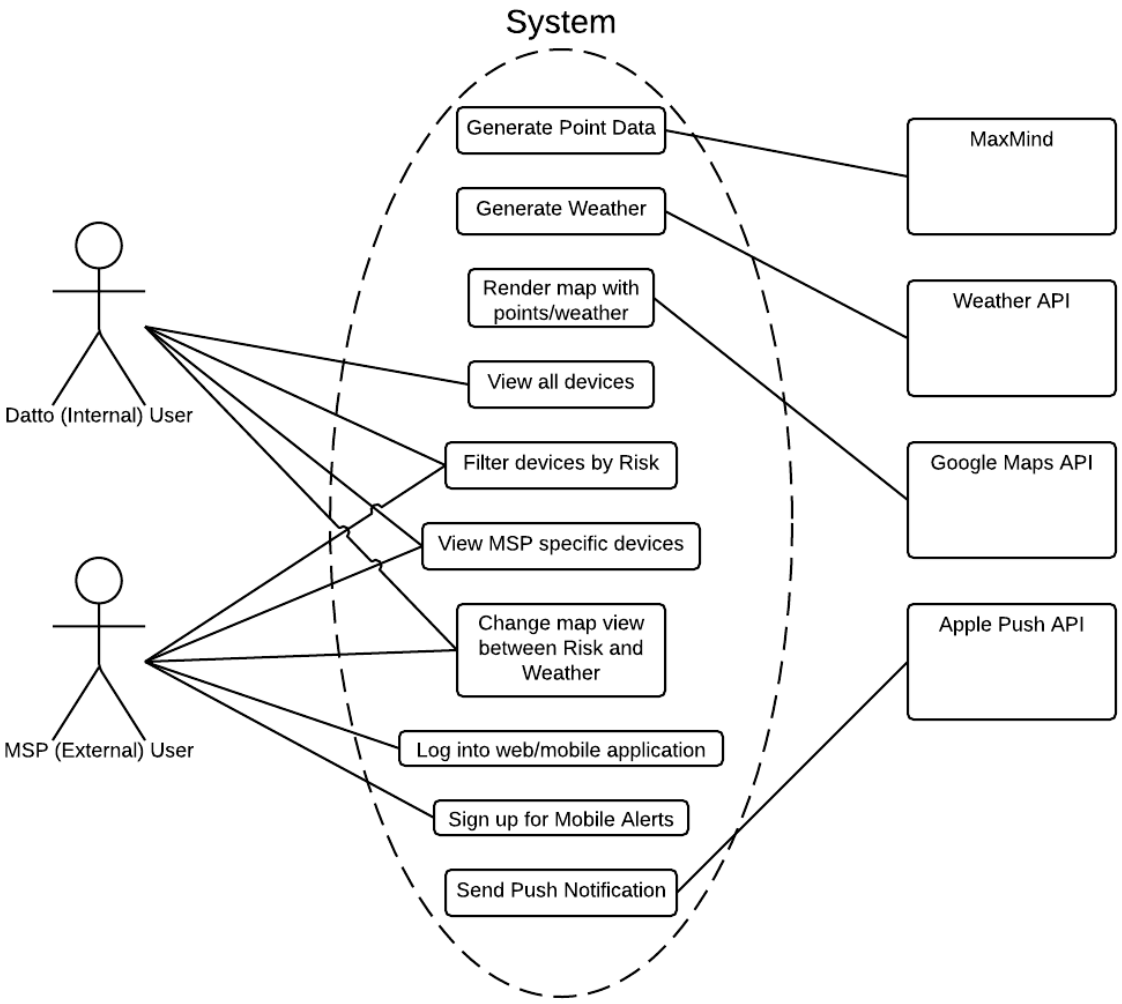
The following diagram details the database schema for our system.

- Devices - The Devices table will act as a cache that stores device information that will be used to populate the risk map. Everyday at 12:00am EST, a job will run that will update the list of devices. Every 10 minutes, another job will run that will update the Risk rating for each device, based on current and upcoming weather patterns around that device's location. This 10 minute interval will coincide with updates the Weather API receives. Notifications will be sent at this time to MSP's whose device's are at risk. The status of sent notifications will also be stored to ensure we do not send multiple notifications for the same device disaster event.
- Cities - At 12:00am EST when the list of devices is being updated each device will also be linked to the Cities table. This will allow us to display relevant location information for devices to users without having to make repeated calls to get the city and state via coordinates. This will cut down on the amount of calls we must make, increasing the performance of the system.
- MSPToAppleDevice - The MSPToAppleDevice table will store user device tokens (Apple_id) that will allow us to send out targeted push notifications to the appropriate MSP when a device becomes at risk.



Use Case Diagram

Basic overview of system functions and actors interacting with the Datto Disaster Tracker.



Generate point data: We retrieve device information from Datto after obtaining the geo-location via the MaxMind API. The points will then be stored within the system database after being instantiated.

Generate weather: Every ten minutes, the system will retrieve weather information using our weather API. Weather information is used to calculate the risk of devices in disaster areas.

Render map with points/weather: Using the Google Maps API, we generate a map of the entire planet, populating it with the various points we received from Datto which represent devices and weather patterns from the weather API.

View all devices: Filter the map so that all Datto devices are present. This is a Datto user interaction only because MSPs will only be provided with their devices.

Filter devices by risk: Changes the way the devices are listed on the side bar. All devices will be ordered from most at risk to least at risk.

View MSP specific devices: An interaction shared between internal and external users of the Datto Disaster Tracker system. Devices from a specific MSP will be displayed on the map instead of all devices. In the case of the external user, this is the only way to view devices on the rendered map.
Change map view between risk and weather: The user of the web application will have the ability to switch between two different views: a risk view that presents numerous points of red, yellow, and green color that represent high, medium and low risk items respectfully; and a weather view which generates a doppler map to give a more accurate representation of weather in the area of the map.
Log into web/mobile application: An external user will log into the system using their Datto MSP credentials.
Sign up for mobile alerts: When logged into the mobile application, an external MSP user can sign up for push notifications to alert them on the status of any devices that are at high risk.
Send push notification: If the mobile external user signs up for mobile alerts and one or more of their appliances is in an at risk area, the apple push notification API will send them an alert to notify them that their devices are at risk.

API Design

- **GET /devices**
  - Input Query Params:
    - msp_id (Optional) - Only show devices from the corresponding MSP.
    - risk_above (Optional) - Only show devices above the given risk value (decimal between 0 and 1).
  - Output Format: (JSON)

```
{
      device_id: String,
      msp_id: String,
      latitude: Float,
      longitude: Float,
      risk: Float,
      city: String,
      state: String
}
```

To Be Defined: Login API