MeshDB Schema Design

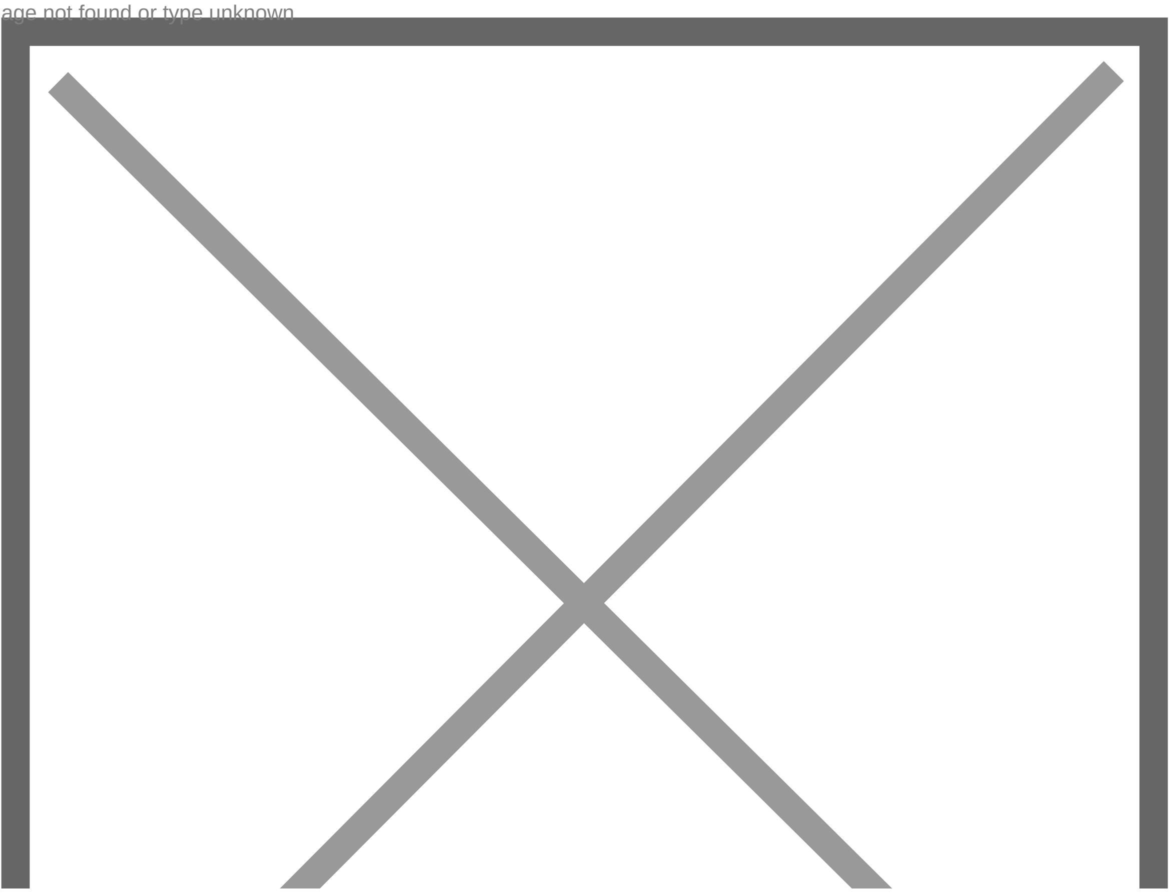
Background

MeshDB is an under-development software application with the goal of replacing the New Node Responses Google Sheet (the spreadsheet) as the source of truth for NYCMesh member, install, geolocation, device, and connection information via a proper SQL database. It is built in the Django ORM, using Python Model objects to represent underlying database schema structures. The schema used for development up to this point is unable to faithfully represent some edge cases that occur at atypical NYC mesh sites. In this document, we propose a modified schema and explain each edge case, detailing how the edge case will be represented under the proposed schema

The Schema (Simplified)

The following diagram depicts the proposed schema, showing the relationships between models (SQL tables), and some key attributes of each model. For clarity, non-essential attributes are omitted (see appendix A for a comprehensive diagram).

Image not found or type unknown



We propose the following models:

1. **Member** - Represents a single NYC Mesh Member (even if they have moved between multiple addresses and therefore have multiple installs or "own" multiple active installs). Tracks their name, email address, and other contact details
2. **Install** - Represents the deployment (or potential deployment) of NYC Mesh connectivity to a single household. This most closely maps to the concept of a row in the spreadsheet. Tracks the unit number of the household, which member lives there, which building the unit is located within. It is keyed by *install number,* which corresponds to row number on the spreadsheet. With foreign keys to Member, Building, and Device, it acts as the central model, tying the entire schema together. Many objects have a status field, but the install status field maps most closely onto the status tracked in the spreadsheet today. Completed Installs have a foreign key to the device field (via_device) which keeps track of the device they use to connect to the mesh
3. **Building** - Represents a location in NYC identified by a single street address (house number and street name). In the case of large physical structures with more than one street address, we will store one Building object for each address that we have received Install requests for. Buildings track a primary network number, to represent the way the site is referred to colloquially. In the case that a building has more than one network number, the primary network number will be set to the one volunteers designate as the “primary” (usually the first assigned, busiest router, etc.)
4. **Device** - Represents a networking device (router, AP, P2P antenna, etc.). Most closely corresponds to a “dot” on the map. Not comprehensive of all devices on the mesh, only those that need a map dot. For big hub sites, this may be only the core router. Contains a mandatory field for “network number” (NN) which will be set to the NN of the device, or of the “first hop” router used by this device (for devices like APs which have no NN assigned). It contains optional lat/lon override fields, which can be used to refine the exact location of this device (e.g. for map display). When no lat/lon are provided for a device, is it assumed to reside at the lat/lon of the building it is

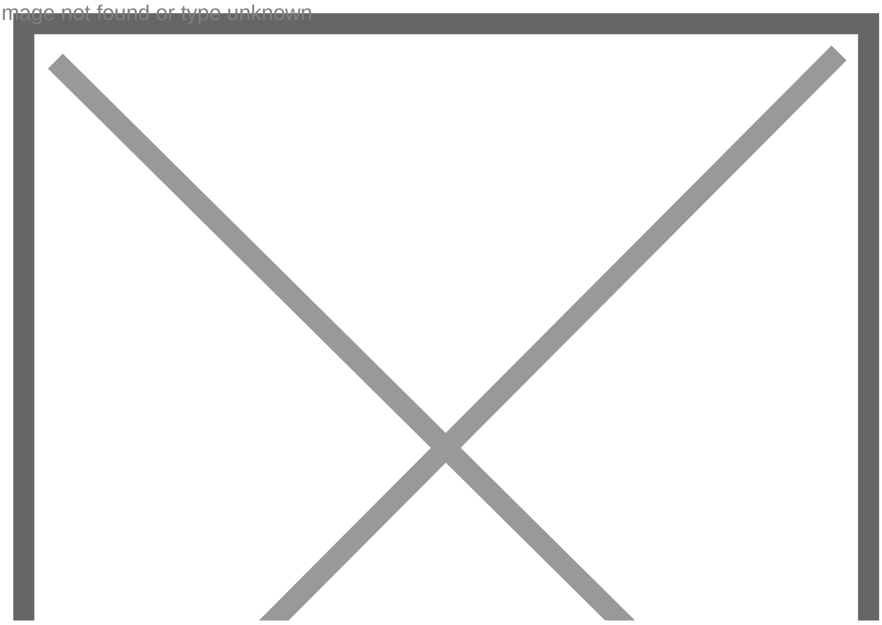
associated with (via the Install model). Devices can optionally track which install delivers them power, via a powered_by_install foreign key to the Install model, which tells us which unit has the PoE injector.

    1. **Sector** - A special type of device (using Django Model Inheritance to inherit all fields from device) which adds additional fields related to the display of sector coverage information on the map (azimuth, width, and radius)

5. **Link** - A connection between devices, which represents a cable or wireless link, whether directly between the devices or via other antennas not represented with their own device objects

Example 1 - NN492 - Typical Multi-Tenant Install

Image not found or type unknown



In this simple example, we have two tenants in a single building with a single address, both connected via cables directly to an omni on their shared roof. They are connected to the rest of the mesh via an LBE to Saratoga. The database tables for this scenario look like this:

Installs		
Install Number	Via Device	Building
13134	1	1
13276	1	1

Buildings			
ID	Primary NN	Address	BIN
1	492	216 Schaefer Street	3079532

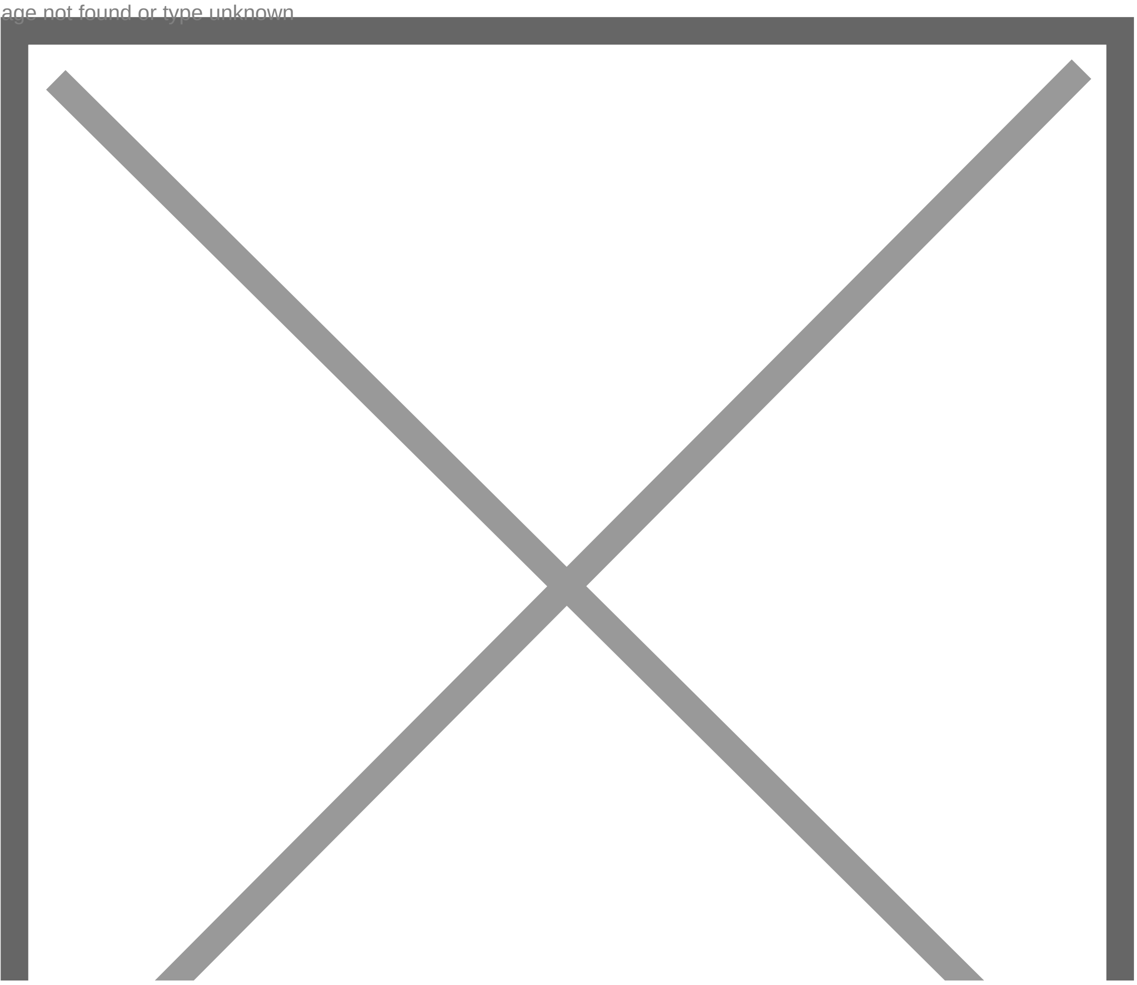
Devices

ID	Network Number	lat/lon overrides
1	492	-

Links		
ID	From Device	To Device
1	1	<saratoga device id>

Example 2 - NN 4734 - Cross-Building Installs

Image not found or type unknown



In this example, members in 3 adjacent buildings, each with their own address, are connected via a single omni, with cable runs across the roofs directly to the member’s apartments. They are connected to the rest of the mesh via an mant 802.11 sector at 4507. The database tables for this scenario look like this:

Installs		
Install Number	Via Device	Building
4734	2	2
6972	2	3
13663	2	4

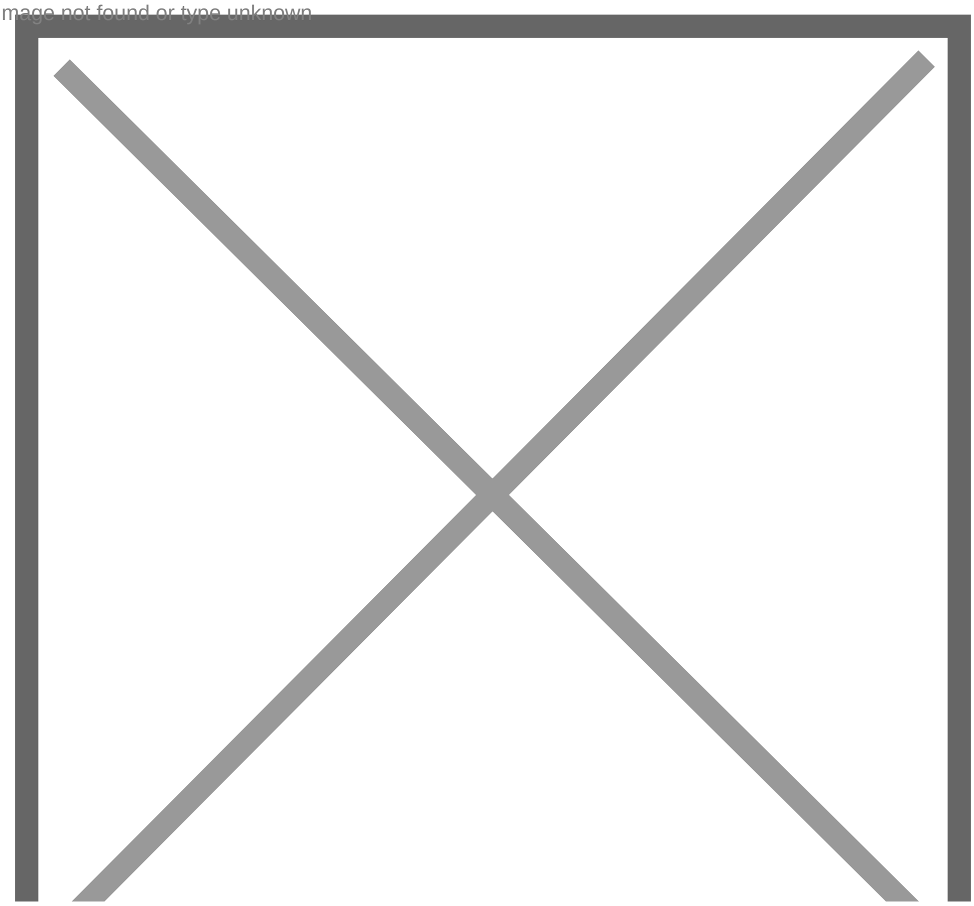
Buildings			
ID	Primary NN	Address	BIN
2	4734	31 Clarkson Ave	3115982
3	4734	25 Clarkson Ave	3115985
4	4734	27 Clarkson Ave	3115984

Devices		
ID	Network Number	lat/lon overrides
2	4734	-

Links		
ID	From Device	To Device
2	2	<4507 device id>

Example 3 - 7th Street (NN 731) - Multiple Omnis on one building

Image not found or type unknown



In this example, we have one regular tenant in a single building with a single address. However there is also a rooftop office with its own omni, connected wirelessly to the primary one. They are connected to the rest of the mesh via a GBELR to Grand. The database tables for this scenario look like this:

Installs		
Install Number	Via Device	Building
731	3	5
12985	4	5

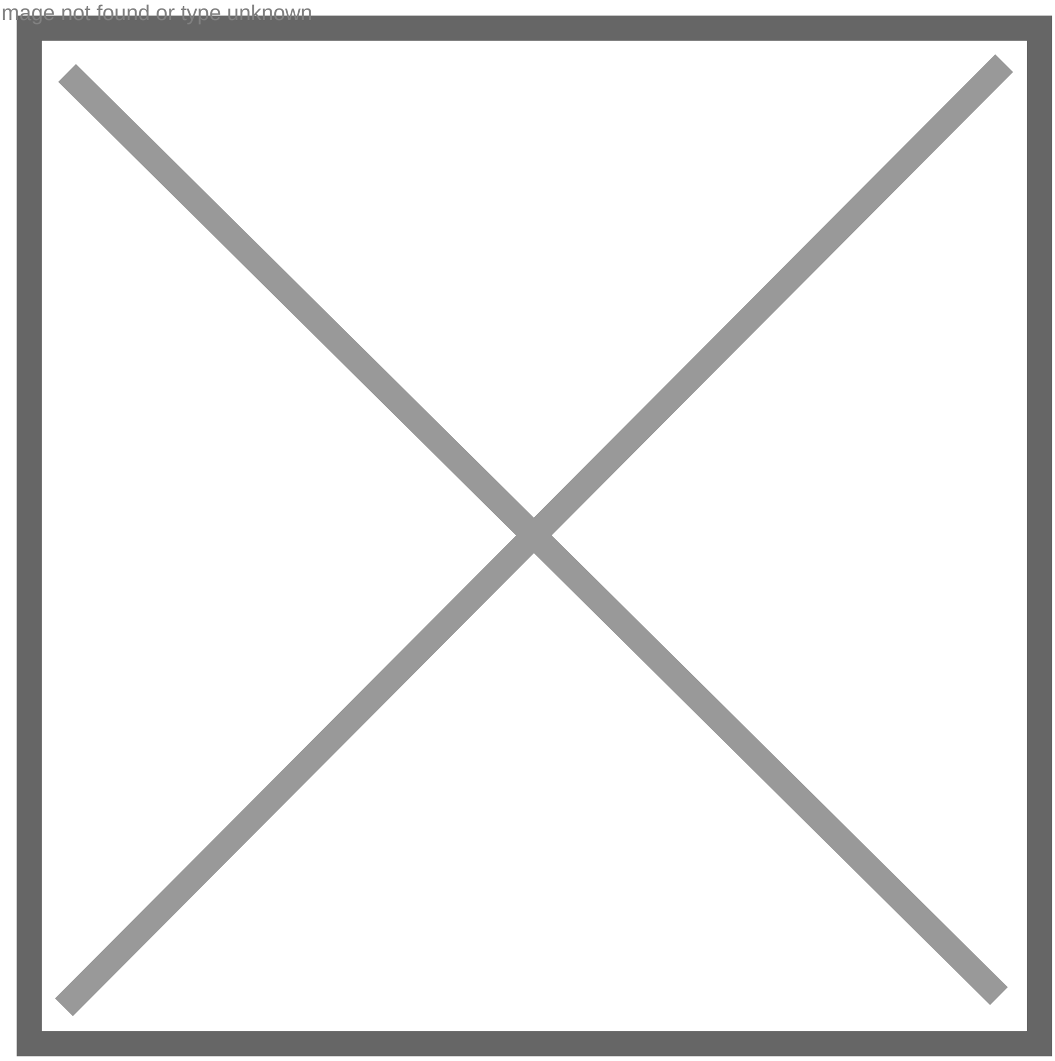
Buildings			
ID	Primary NN	Address	BIN
5	731	190 East 7th Street	1086499

Devices		
ID	Network Number	lat/lon overrides
3	731	-
4	311	x, y

Links		
ID	From Device	To Device
3	3	4
4	3	<1932 device id>

Example 4 - Vernon (NN 5916) - Courtyard APs

Image not found or type unknown



In this example, we have a core hub site in a single building with a single address. However, there are many Access Points (APs) on light poles in the building’s courtyard. These light-poles are unquestionably associated with the same building/address as the core router of this hub, but need to be shown separately on the map.

In this scenario, we treat the light poles as if they are “apartments” in the Vernon building. They each get their own install #, but imagining a tenant living in the light pole, we say that this imaginary install is “connected via” a device object representing the AP. The network number for these APs is set to 5916, reflecting their first hop router (and the fact they are not themselves assigned NNs). Links between the courtyard APs and the core router are included so that they are rendered on the map

The database tables for this scenario look like this:

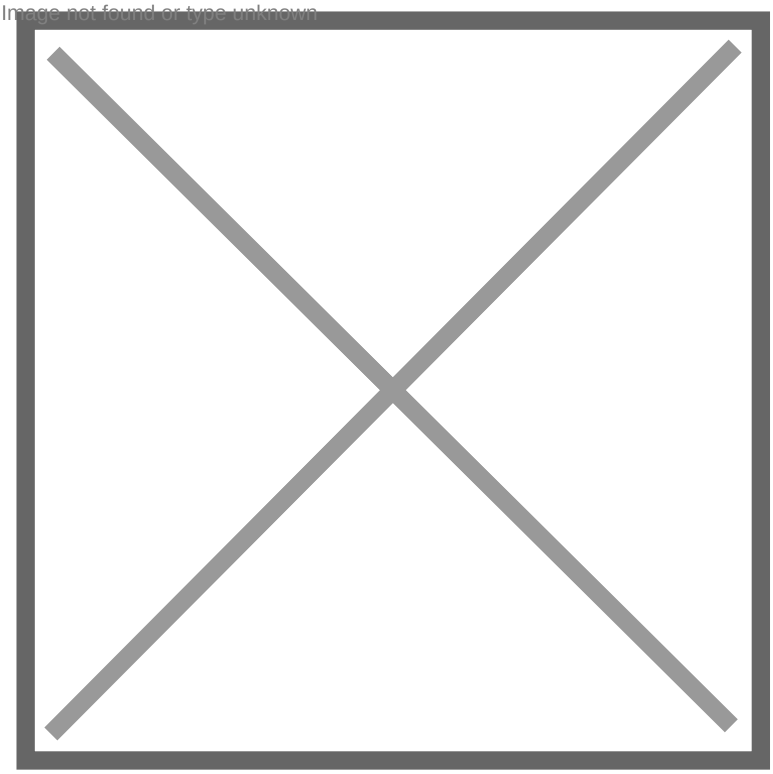
Installs		
Install Number	Via Device	Building
5916	5	6
6345	-	6
11875	6	6
11876	7	6
11877	8	6
11878	9	6
11879	10	6
11880	11	6

Buildings			
ID	Primary NN	Address	BIN
6	5916	303 Vernon Avenue	3042881

Devices		
ID	Network Number	lat/lon overrides
5	5916	-
6	5916	x, y
7	5916	x, y
8	5916	x, y
9	5916	x, y
10	5916	x, y
11	5916	x, y

Links		
ID	From Device	To Device
5	5	<SN3 device id>
6	5	<grand device id>
7	8	9
8	8	6
9	6	5

Example 5 - Prospect Heights (NN 3461) - Multiple NNs for one building



In this example, we have a core hub site in a single building with a single address. The primary NN 3461, also serves a member's apartment as install #3461. However, there is another apartment which could not due to practical considerations be connected via a cable, and had to be connected via an antenna in their window to a sector on the roof. This antenna needed an NN for configen and naming, and so this building received multiple NNs.

The database tables for this scenario look like this:

Installs		
Install Number	Via Device	Building
3461	3461	7
3921	-	7
6723	-	7
11024	377	7
14399	-	7
14960	-	7

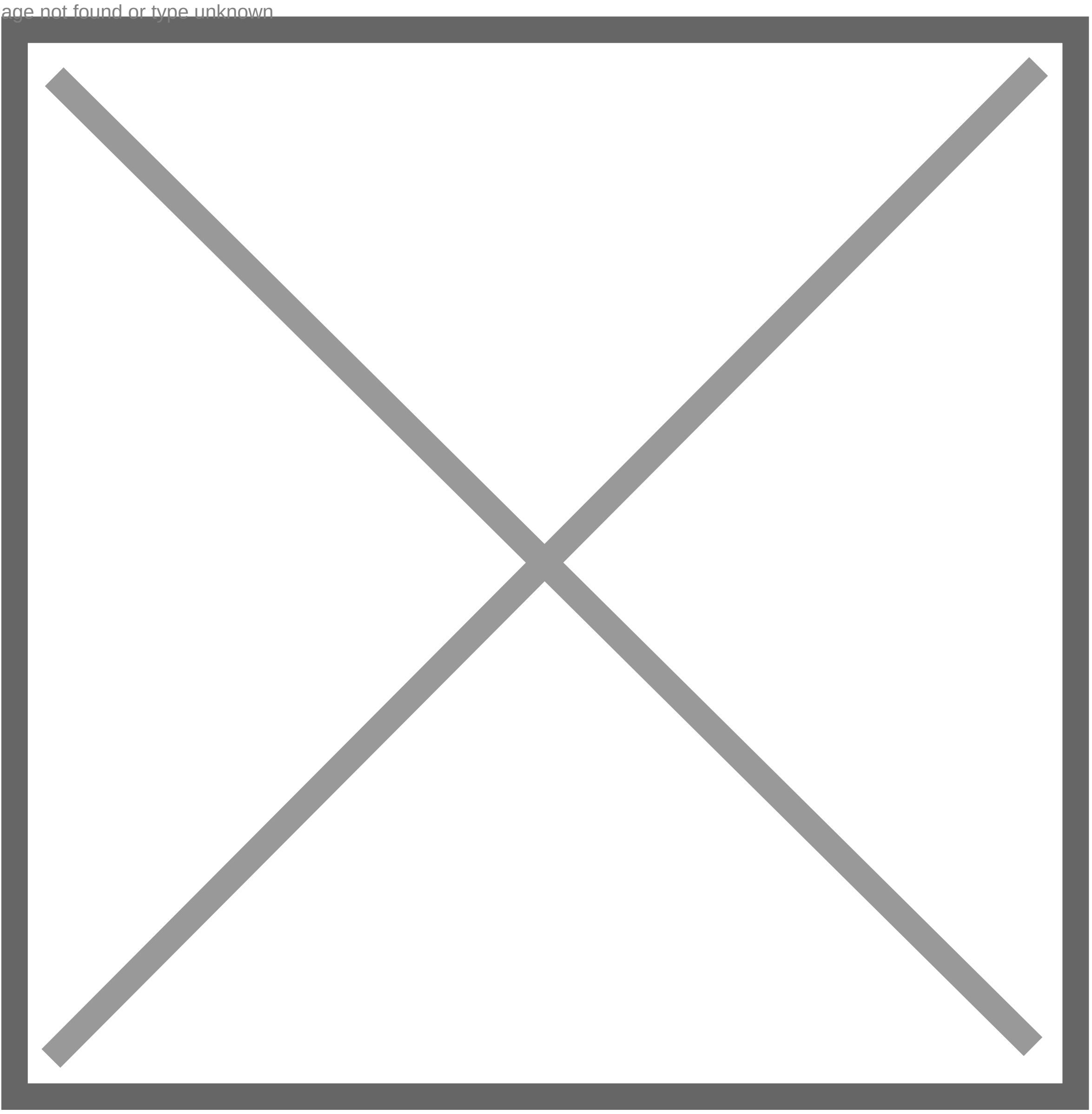
Buildings			
ID	Primary NN	Address	BIN
7	3461	135 Eastern Parkway	3029628

Devices		
ID	Network Number	lat/lon overrides
12	3461	-
13	377	x, y

Links		
ID	From Device	To Device
10	12	<SN3 device id>
11	13	12

Example 6 - Jefferson (NN 3606) - Multiple NNs for multiple buildings

Image not found or type unknown



In this example, we have a building with 4 addresses and 3 omnis on the roof, each with its own network number. There is no clean mapping between NNs and addresses, since each omni serves installs in multiple buildings. The omni of the primary NN, 3606, provides the uplink to Hex House (NN 1417).

The database tables for this scenario look like this:

Installs (omitting abandoned & potential for brevity)		
Install Number	Via Device	Building
3606	14	8
5933	15	8
7177	15	8
8152	16	8
8274	14	9
8085	16	11

Buildings			
ID	Primary NN	Address	BIN
8	3606	476 Jefferson Street	3819572
9	3606	488 Jefferson Street	3819572
10	3606	28 Scott Avenue	3819572
11	3606	16 Cypress Avenue	3819572

Devices		
ID	Network Number	lat/lon overrides
14	3606	x, y
15	5933	x, y
16	169	x, y

Links		
ID	From Device	To Device
12	14	<1417 device id>
13	14	15
14	15	16
15	16	14

Appendix A - Full Schema Diagram

The following is a complete schema diagram, showing all fields. New additions from the current implementation are shown in yellow, and removed fields are shown in red

Image not found or type unknown

Revision #8
Created 23 February 2024 03:55:10 by Andrew Dickinson
Updated 29 February 2024 06:53:21 by Olivier