

# Deploying Google's Enterprise WiFi

...in TEN minutes.

Mike Albano  
@mike\_albano  
[albanom@google.com](mailto:albanom@google.com)  
WLPC Phoenix 2019

# Agenda

1. How we used to do it.  
(Why that sucked.)
2. How we do it now.  
(Why it sucks less.)

# 7+ Years Ago...



- Build rooms.
- Docs w/ “standard” configs.
- 400+ employees with Read/Write access.
- Vendor NMS’ for monitoring + OSS (RANCID etc.)
- Dynamic tx-power & channel assignment.
- About 0 automation.

Entirely vendor-specific and institutional knowledge required to deploy, monitor & troubleshoot.

We used 1 vendor.

# 4 Years Ago...



- Build rooms.
- ~~Does w/ "standard" configs.~~
- Config Generators.
- ~~Vendor NMS' for monitoring.~~
- Expect Scripts (CLI).
  - LOTS of scripts.
- SNMPGets (slow).
- ~~Dynamic tx power.~~
- Static Tx Power(s).
- Dynamic Channels.
- Some automation.

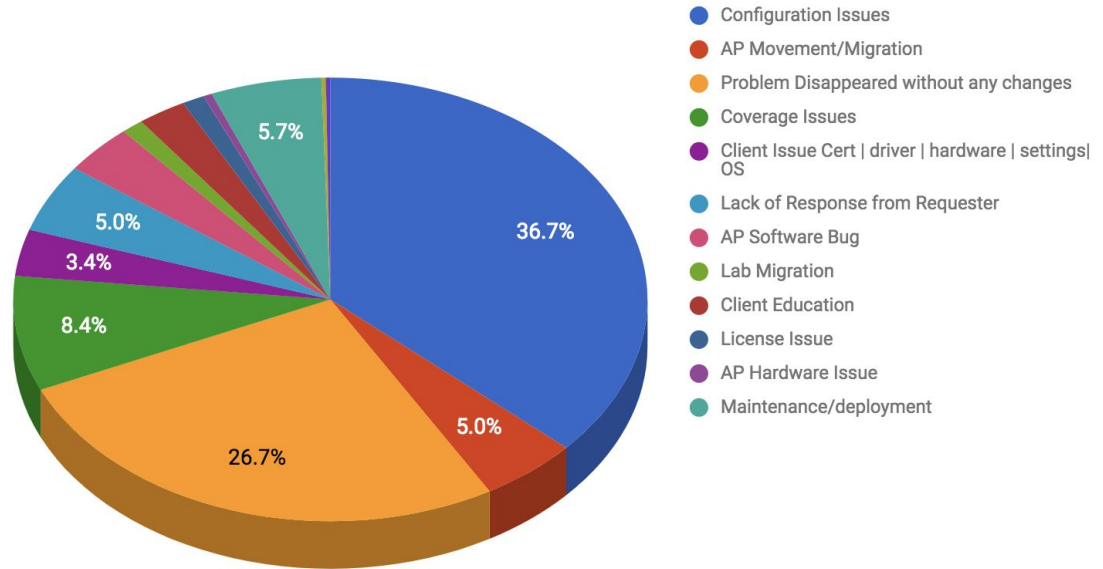
Entirely vendor-specific and institutional knowledge required to deploy, monitor & troubleshoot.  
We used 1 vendor.

# 500 humans, 1000's of buildings, 600 controllers; all globally dispersed leads to...

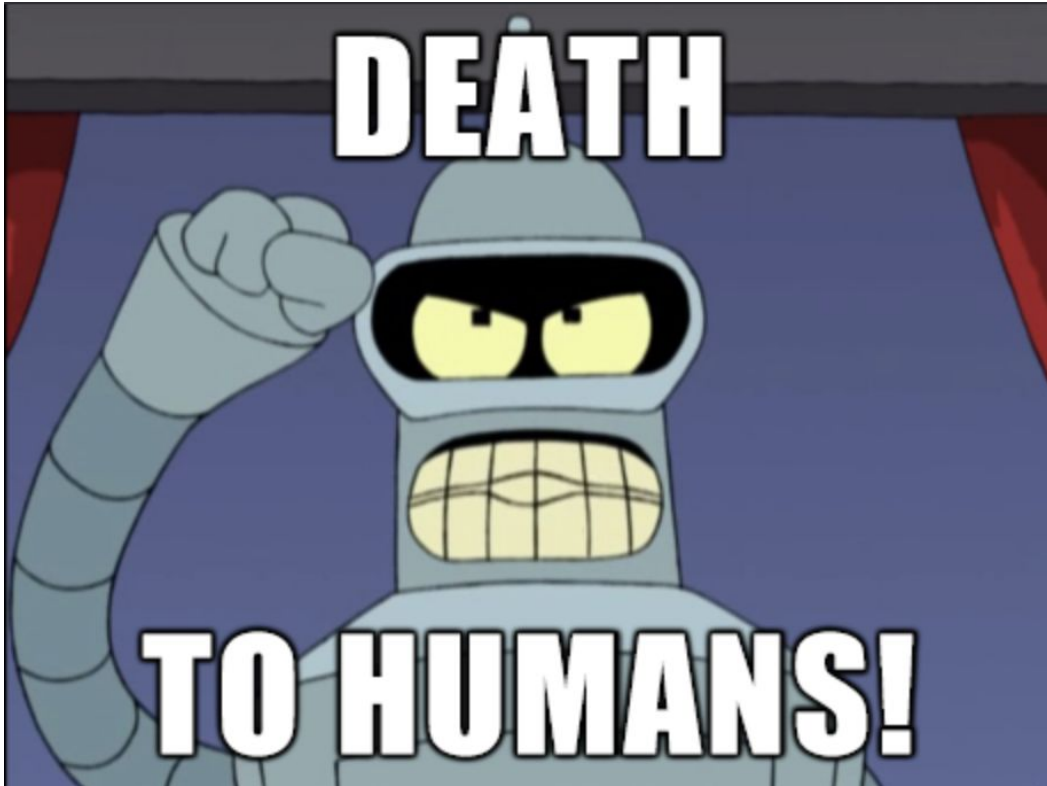
- Over 1/3 of problems stem from configuration issues.
- Fault-isolating based on “user reports” is itself a BIG problem.

Note, “AP Software Bug” is at times MUCH higher cause; but we often spot prior to user-reports.

Root Cause



What did we want to change?



# 1000' view of realizations...

1. We need Telemetry (radio-data) and we need it fast.
2. We need multi-vendor, without translation layers; freedom of choice.
3. We need programmatic access and structured APIs for everything.
4. We need absolute predictable configuration.

# But there's always humans

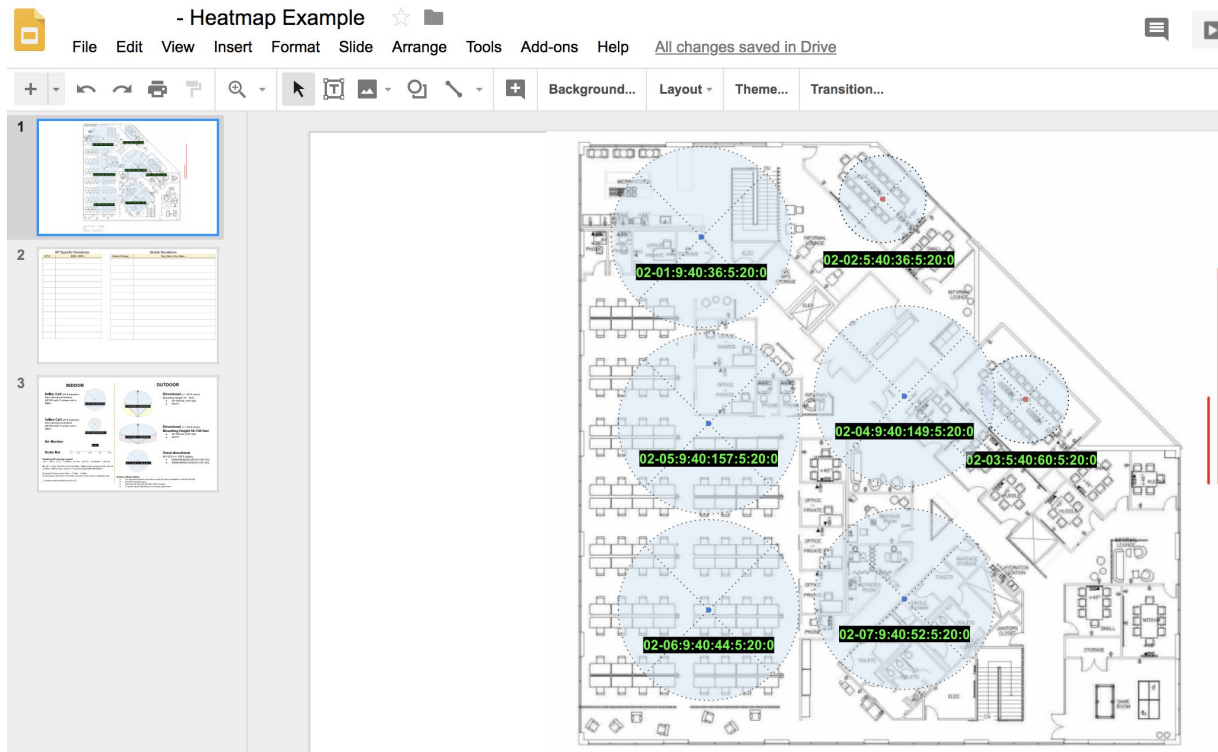
- One thing that humans SHOULD ALWAYS do: **Generate Heatmaps**.
  - Necessary to establish AP placement source-of-truth.
  - Required for troubleshooting (WiFi problems are often situational).
  - ...and yet, somehow never accurate?
- Intent, just like configuration and documentation, is **ONLY as accurate as it HAS TO be.**

This is why we made the **Heatmap ==** our **ONLY** intent resource from which **ALL** configuration is derived.



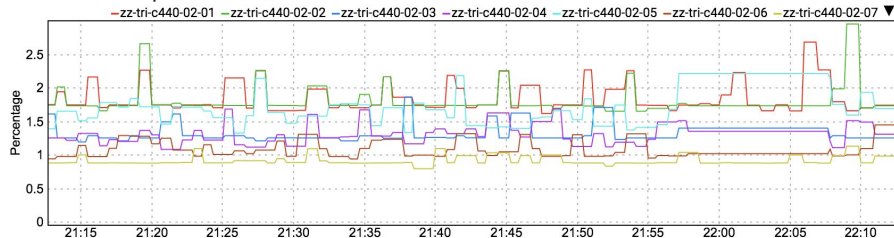
# [Deploy] Example

- Slides API (Get PHY config & placement checks).
- Sheets API to populate a gSheet, used by scanner App.
- Drive API for locating & moving files.
- IPAM API Hostname/IP registration.
- gNMI/OpenConfig API for config/telemetry (multi-vendor)
- <favorite\_RCS> API for revision control of final OpenConfig JSON.

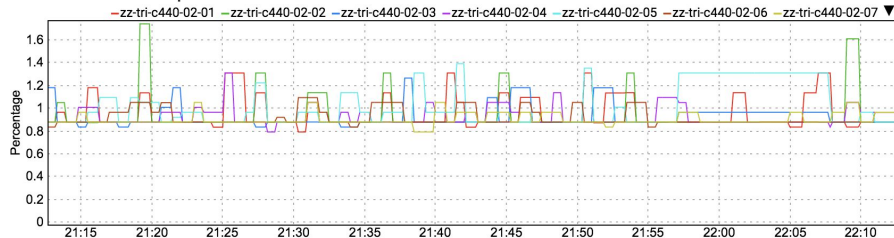


# [Operations] Example

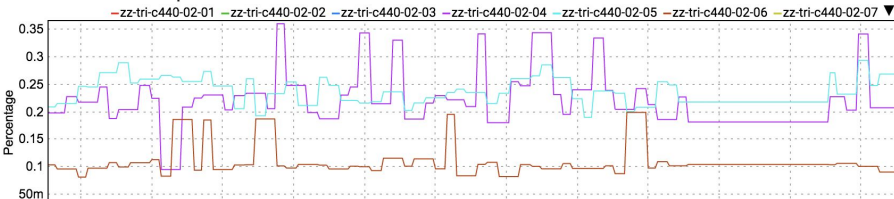
Total Channel Utilization per Host Name



Tx Dot11 Channel Utilization per Host Name



Rx Noise Channel Utilization per Host Name



- Yea, we graph stuff.
- Not revolutionary, but we DO rely 100% on Telemetry exposed via OC API.
- This is ONLY insight into network.
- One “pane of glass”, regardless of vendor in use.

# What this gets us...

- The **ONLY** way to add/move/change APs is via Heatmap.
- After physical mounting; one script provisions, configures, channel plans.
  - 1 Hour Vs. 20+
- **ALL** vendors deployed, configured AND monitored in exact same way.  
Tier 1, 2 & 3 do not know what vendor is in use.
- No build rooms/prepping equipment.
- No “golden configuration” docs; PHY-layer info is where it belongs -- the Heatmap, MAC-layer in code.
- Scanner App reads names from Heatmap & Updates MAC <> Hostname binding.
- **0** vendor **proprietary** tools.
- RW to vendor native interface exposed to small team (5 humans).

# If you haven't run out of time, say this to get people to argue with you...

We run **static** channels & transmit-power, **everywhere**, all the time. It's scalable, easy, predictable and **better in every way** in our environments; which are limited to:

- Office buildings
- Datacenters
- Warehouses
- Stripmalls
- Broom Closets
- Cafe's
- Classroom Buildings
- Outdoors
- Mike's House
- Medical facilities
- Manufacturing Plants