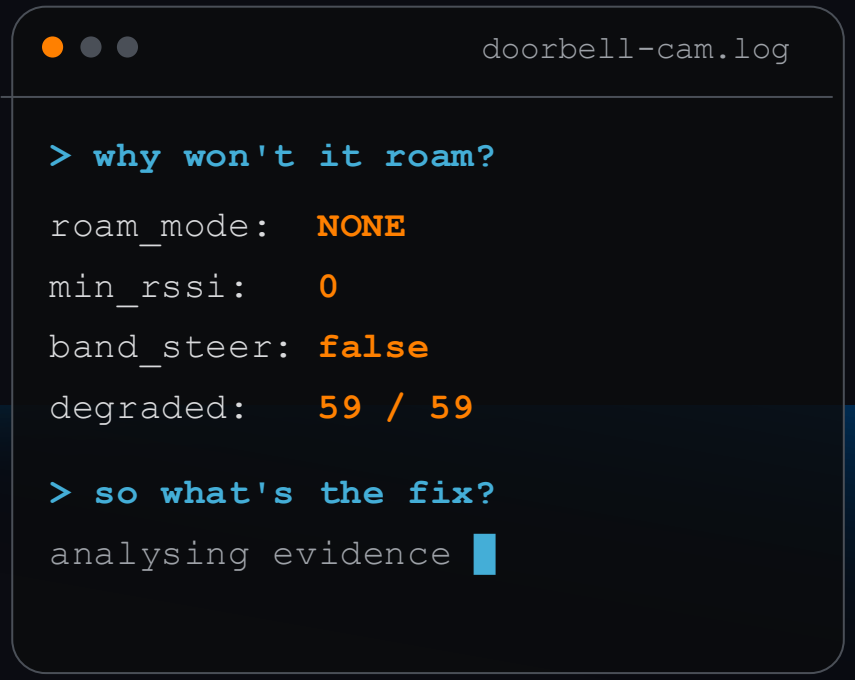
Effectively Using LLMs to Troubleshoot Wi-Fi

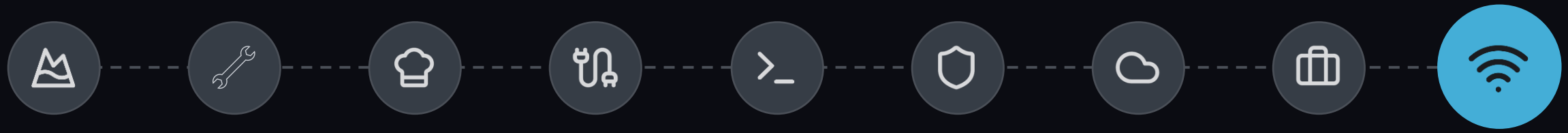


Or: how the evidence made me a better engineer

Rory Standage · WLPC

A quick introduction

Rory Standage · Principal Engineer, HPE Mist Field PLM



WHAT I BROUGHT

- Junos switching & security
- Network automation
- Cloud architecture

THE PLAN, FOUR YEARS AGO

"I'll pick Wi-Fi up as I go."

Expectation in this team: Everyone thinks I already know Wi-Fi 😳

So I leant into this community

1000s

of years of collective knowledge

Debated, shared and presented to better everyone. Nothing like it anywhere else in IT.

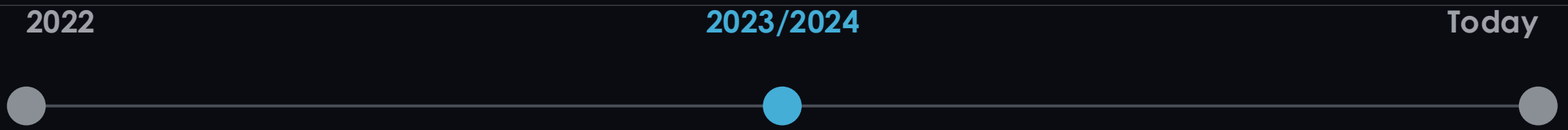
Speed dial

- Rob
- Coops
- Phil Keeley

I learn by solving problems

CCNA × 4
failed attempts

Also: quite lazy. So automation has always been a passion.



How can we effectively use LLMs to troubleshoot Wi-Fi?

or, more to the point…

How can I use AI tools to make me a **better engineer?**

PART 1

Where it goes wrong

Great at method. Bad at facts.

Great at

- Pattern recognition
- Anomaly detection
- Executing a method
- Reasoning over data

Not great at

- Nuance
- Facts & references
- Your domain, your network
- Knowing when it's unsure

Faster than us at crunching data. **Worse than us at knowing when it's wrong.**

Failure #1 — The invented reference

Is there a known bug for clients dropping right after a fast roam on this firmware?

Yes — this is a known issue tracked as **PR 1734291**, resolved in **11.2.4-S3**. Upgrading will fix the disconnects; no config change is needed.

DOES NOT EXIST

Confident.

Specific.

Fictional.

Irrelevant.

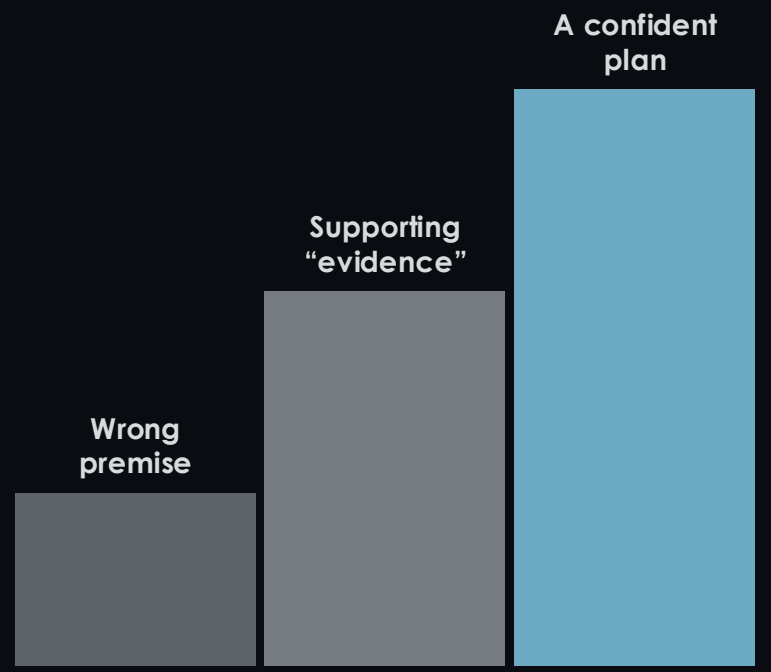


Failure #2 — The sycophant

I reckon it's DFS — radar events kicking clients off the channel.

Great catch — that fits perfectly. DFS explains the drops, the timing and the affected APs. Next, let's build a channel plan that avoids…

More work to disprove



Feels like progress. **Isn't.**

PART 2

Making it useful

"A raw LLM is a very confident post-grad who has never seen your network — and read all the technical books a year ago."



FIX 1

Skills

Give it a method



FIX 2

Live data

Let it see the network



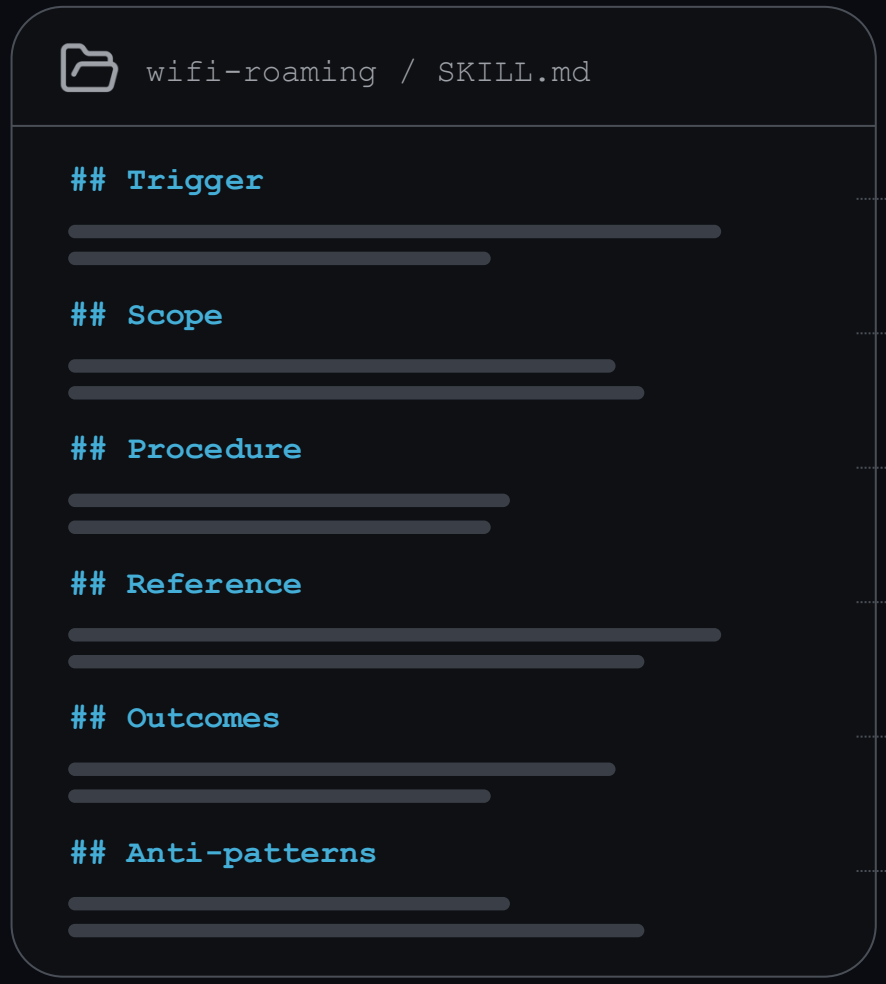
FIX 3

Memory

Let it remember

Fix 1 — Skills: the method, written down

Stop asking what it knows. Tell it what to do — and where to look.



1. What kicks off this skill
2. What it does/doesn't cover
3. The METHOD
4. Reference data
5. Good/Bad Examples
6. What not to conclude

What a skill actually looks like



JunOS Expert — Juniper CLI & configuration

Authoritative knowledge for how to configure and inspect Juniper JunOS devices (EX/SRX/MX/QFX). Troubleshooter skills consult this for the exact commands and config syntax. Pure reference — no workflows. For *what the output means* (protocol behavior) see `route-switch-expert`; for Mist platform/cloud management see `mist-expert`. **SSR (Session Smart Router) is not JunOS — out of scope here.**

Every command and config snippet in this skill is taken from vendor documentation or on-box output — never from memory or reasoning. Do not trust a JunOS command, flag, or path you cannot point to a source for: see the **Evidence & accuracy rules** below.

QFX switches (QFX5120, QFX5130, QFX10002/10008, QFX5200) run JunOS and appear in campus deployments as collapsed-core switches or small EVPN-VXLAN fabrics.

Quick reference

- **Config model** (`references/config-model.md`): candidate config, `set` / `delete`, `show | compare`, `commit` / `commit confirmed` / `rollback`, groups & apply-groups, ELS vs non-ELS detection, config format intake.
- **Operational commands** (`references/operational-commands.md`): the diagnostic `show` catalog — interfaces/optics, `show route`, `show ethernet-switching table`, OSPF/BGP, LLDP, PoE, chassis.
- **EX** (`references/platform-ex.md`): ethernet-switching (access/trunk/native-vlan), irb, Virtual Chassis, PoE, storm-control, MTU, firewall filters, common misconfigs.
- **SRX** (`references/platform-srx.md`): zones, security policies, flow/session, NAT, chassis cluster, tcp-mss, screens, common misconfigs, SRX log signatures.
- **QFX** (`references/platform-qfx.md`): campus collapsed-core / small EVPN-VXLAN fabric — models, EVPN-VXLAN config, MC-LAG, breakout, common misconfigs.
- **Monitoring/logging** (`references/monitoring-logging.md`): syslog primer, daemon glossary, which-logs-to-pull guide, `show log`, `traceoptions`, `monitor traffic/interface`, RSI.

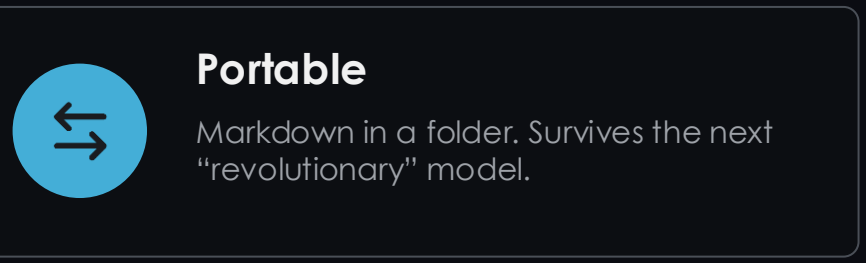
Reference index

- `references/config-model.md` — JunOS config model & safe-change workflow; ELS detection; config-format intake
- `references/operational-commands.md` — diagnostic show-command catalog
- `references/platform-ex.md` — EX switching specifics (access/trunk, IRB, VC, PoE, storm-control, MTU, filters, misconfigs)
- `references/platform-srx.md` — SRX security specifics (zones, policy, NAT, cluster, tcp-mss, screens, log signatures, misconfigs)
- `references/platform-qfx.md` — QFX campus collapsed-core/small fabric (EVPN-VXLAN, MC-LAG, breakout, misconfigs)
- `references/monitoring-logging.md` — on-box diagnostics & evidence gathering (syslog primer, daemon glossary, log sources)

Evidence & accuracy rules

JunOS is exact: a misplaced flag, a wrong hierarchy path, or a platform-only verb turns a fix into an outage. This skill is the authoritative reference precisely *because* its commands are sourced, not guessed. The rules below apply both to keeping this skill correct and to using it.







The ceiling is the engineer who wrote it — and the engineer who checks it.

Fix 2 — Live data. But what leaves the building?

- Hostnames
- 802.1X usernames
- MAC addresses
- Floorplans & AP placement

Not router uptime counters.

A record of **where people were**, and what they connected to.

- Which fields?
- Going where?
- Under what contract?
- Kept how long?

Same technology. Three risk profiles.



Local model

On your own hardware

DATA RISK

Sends nothing anywhere.



Enterprise API / walled garden

e.g. a cloud AI platform tier

DATA RISK

A processing agreement with retention terms you can actually read.



Personal subscription

Claude Pro on your personal email

DATA RISK

Consumer terms. And where a lot of people start.

Know who you're giving your data to.

Stop pasting. Start fetching.

Paste a log

- ✓ A snapshot you chose
- ✓ Easy to sanitise first
- ✗ Manual, one-shot
- ✗ Only as good as your guess at where to look

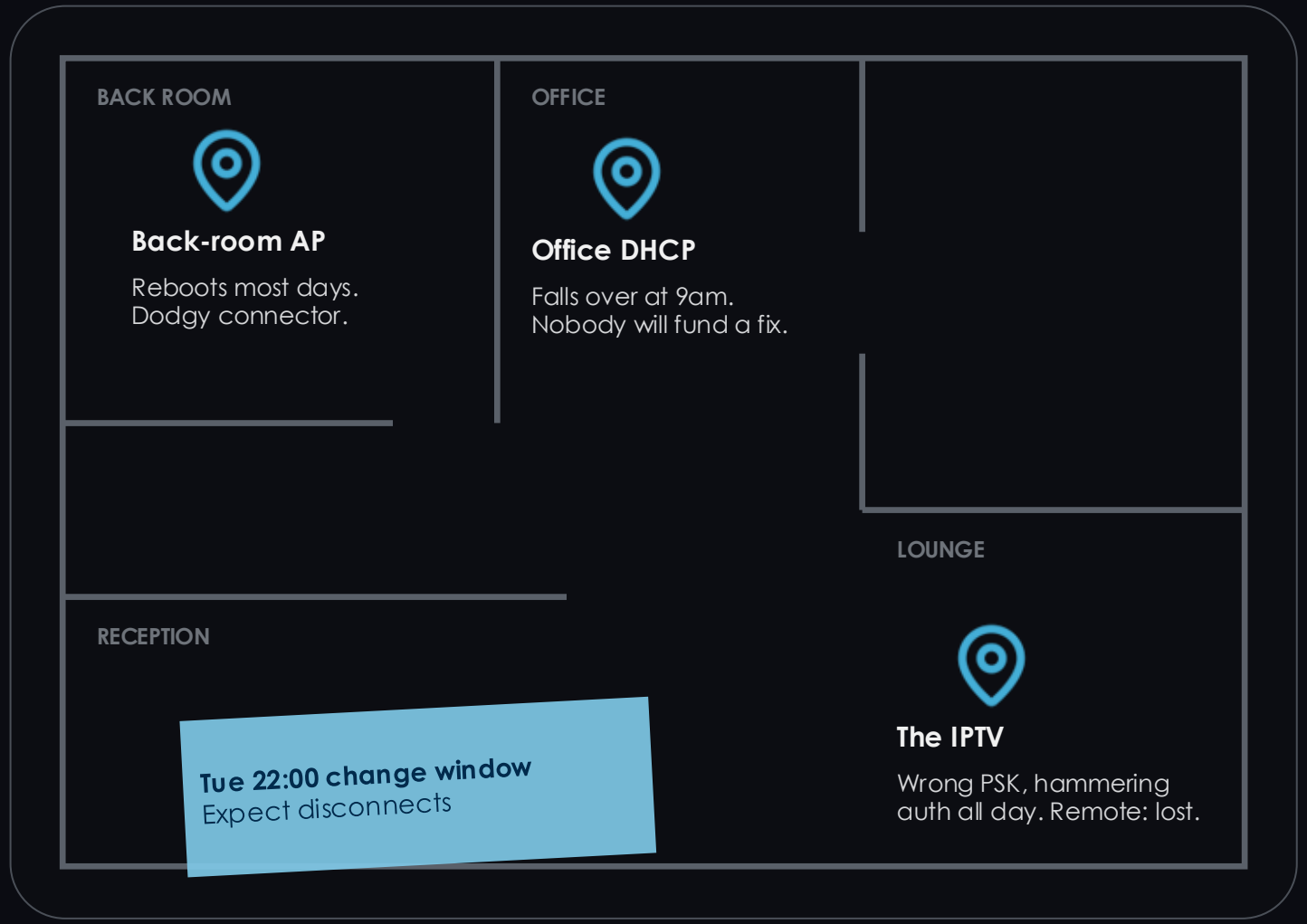
Tool access (MCP)

It queries again based on what came back.



One protocol, not one integration per vendor — your skills and context move with you when the model changes.

Fix 3 — Memory: stop re-explaining your network



Bank what only a few people know.

It's yours — it follows you across models.

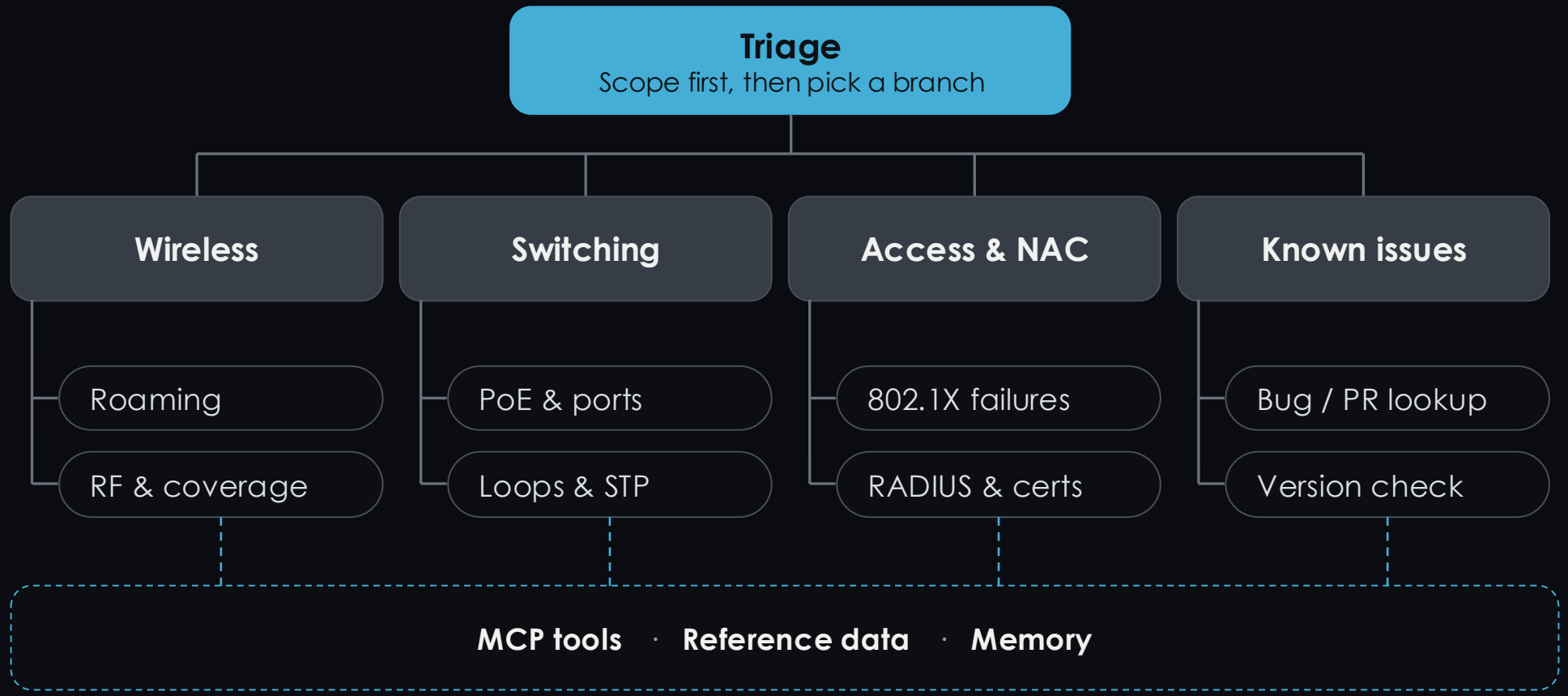
A hierarchical skill tree

ROUTER
Always loaded

DOMAINS
Loaded on route

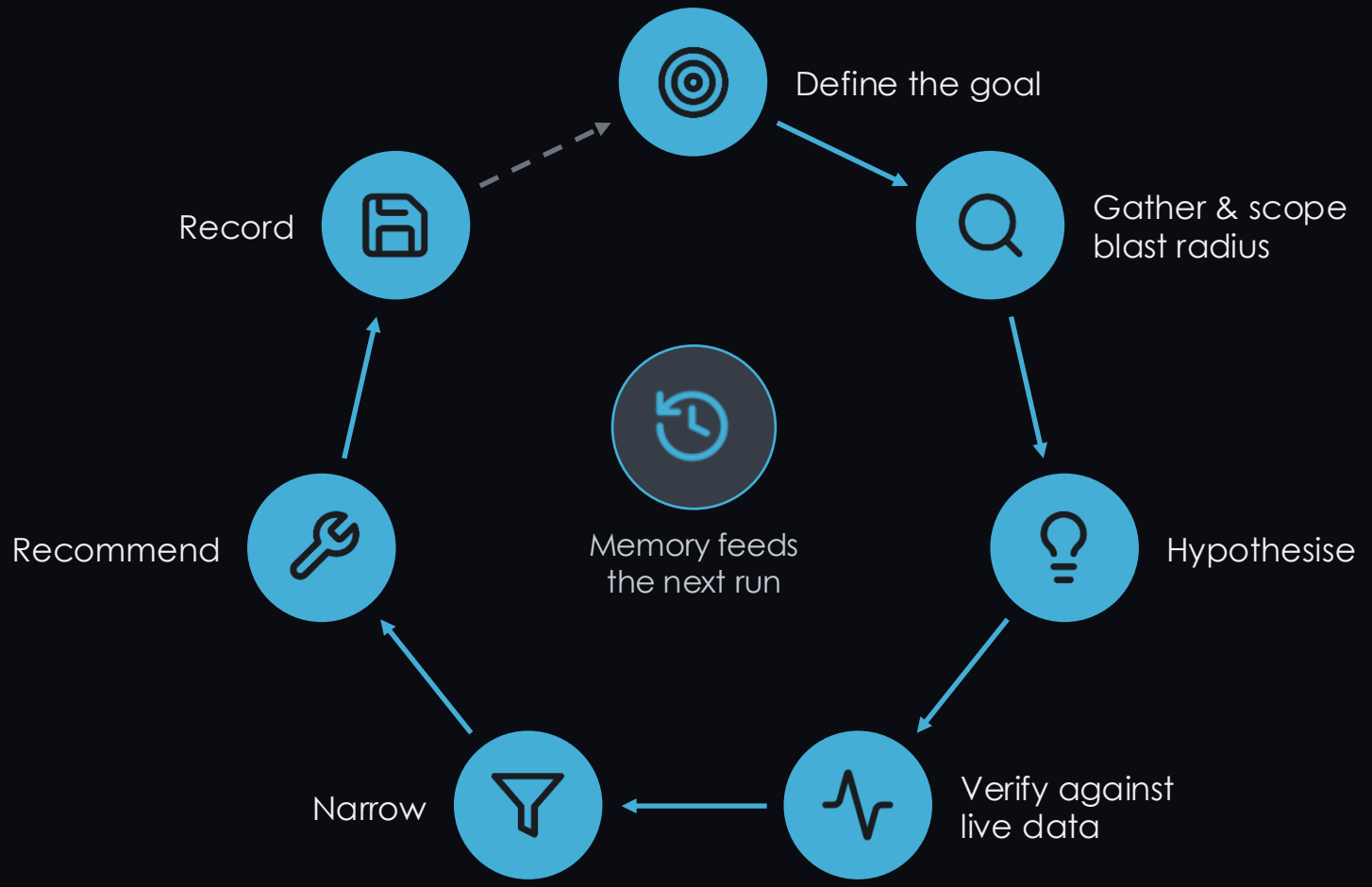
PROCEDURES
Experts loaded on demand

TOOLS & DATA
Pulled by the leaf



Each question walks **one path** through the tree, not your whole skill library.

The agent loop



PART 3

What it taught me

Case study: hunting the worst roaming client

One prompt to my agent: which client on my home network roams worst?

IOT FLEET · MIST-WIFI

23 / 23

degraded roams on a Nest Mini that never moves

Cell-edge artefact

IPHONE · MIST-AA

−89 dBm

roam decisions during an 18-roam storm in under 4 minutes

Looked bad. Roamed fine.

IPHONE · MIST-6GHZ

1,507

sessions in 7 days, barely flagged by the roaming score

The real problem

The obvious suspects were noise.
The worst client looked healthy.

Good signal, fast roams on paper, 50 failures underneath

Wireless client event log — 7 days

```
CLIENT_REASSOCIATION_FAILURE

status_code: 53

reason: Invalid PMKID

rssi: -49 to -57 dBm

roam_mode: NONE
```

REASSOCIATION FAILURES

50

in 7 days, every one status 53

WHAT THE ROAMING SCORE SAW

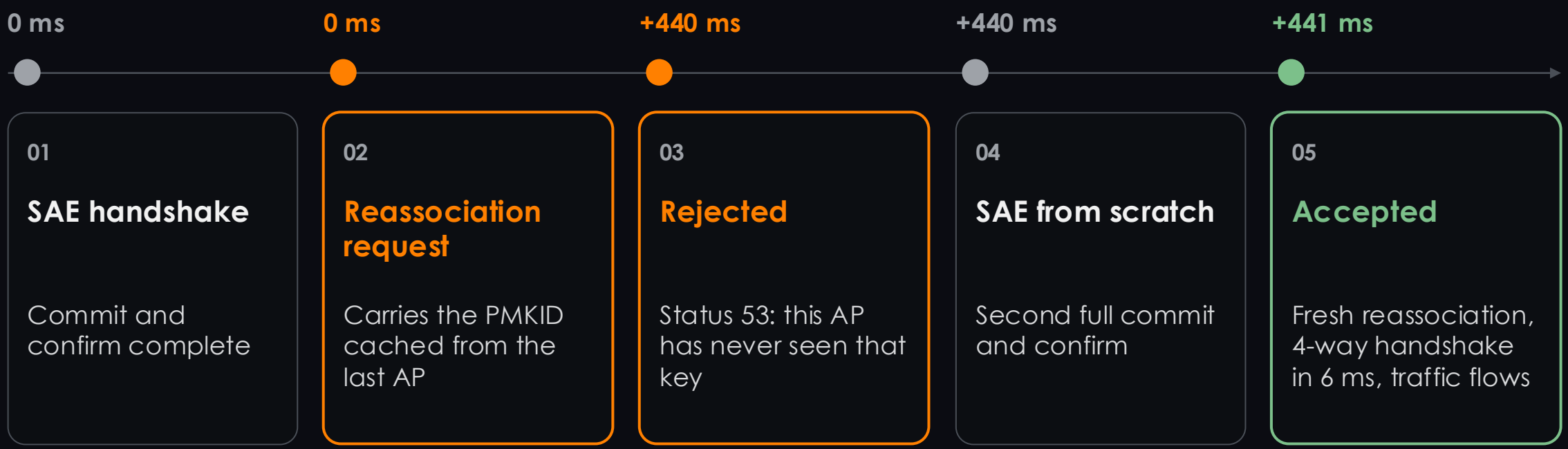
134 / 135

roams completed in under 1 s

Strong signal, zero interference, **and the AP still said no.**

The capture: a 440 ms stall, then a full restart

Marvis captured the failure automatically: 35 frames, 0.44 seconds, AP_03 on 6 GHz.



The fast path failed every time.

The slow path quietly covered for it.

The same iPhones roam in 10 ms when the SSID lets them

Same network, same iPhone platform. The only variable is the SSID config.

MIST-AA · EAP-TLS

`roam_mode: 11r`

~10 ms

fast transition reassociation, every roam

MIST-6GHZ · WPA3-SAE

`roam_mode: NONE`

~0.5 s

per failed roam: PMKID rejected, full SAE restart

The fix: **roam_mode OKC, or FT-SAE (11r)**

Also: max_idletime of 1,800 s forced 26 idle reconnects a week, each one another PMKID rejection.

What I took to the speed dial

Three questions I wouldn't have asked before this session.

01	Does OKC work with SAE on our firmware, or is FT-SAE the only option on 6 GHz?	DESIGN
02	Why does iOS keep leading with a cached PMKID after 50 rejections?	CLIENT BEHAVIOUR
03	Should a roaming score count failed fast-roam attempts, not just roam latency?	MEASUREMENT

MONDAY MORNING



Write down the thing you explain to juniors most — as a skill. Twenty minutes.



Point it at live data before you trust a word it says.

"The skills that would help us most are the ones **only people in this room could write.**"

Thank you · Rory Standage