



Memfault

How to Build and Maintain IoT Management Systems for Scale

Chris Coleman - Co-Founder & CTO
Tyler Hoffman - Co-Founder & Head of Developer
Experience

Chris Coleman & Tyler Hoffman

Co-Founders of Memfault

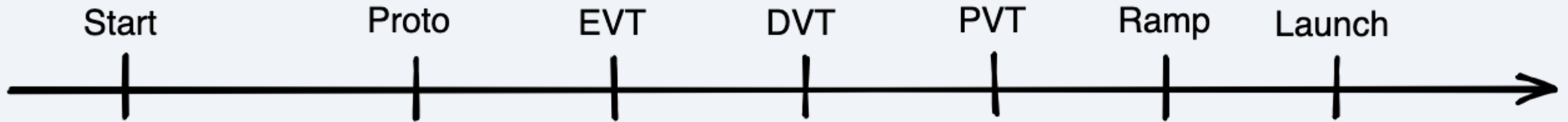
- Passions: tooling and automation in firmware engineering
- Previously Firmware Engineers @ Pebble and Fitbit
- Can find their thoughts and content on Memfault's Interrupt blog (interrupt.memfault.com)



pebble.



A Previous Webinar by François



Launching an IoT Device

[Watch: Launching an IoT Device - A Blueprint for Success](#)

Today's Webinar



Scaling an IoT Device

Four Topics

**Firmware
Team**



**Debugging in
Production**



Scaling Data



**Usable
Systems**

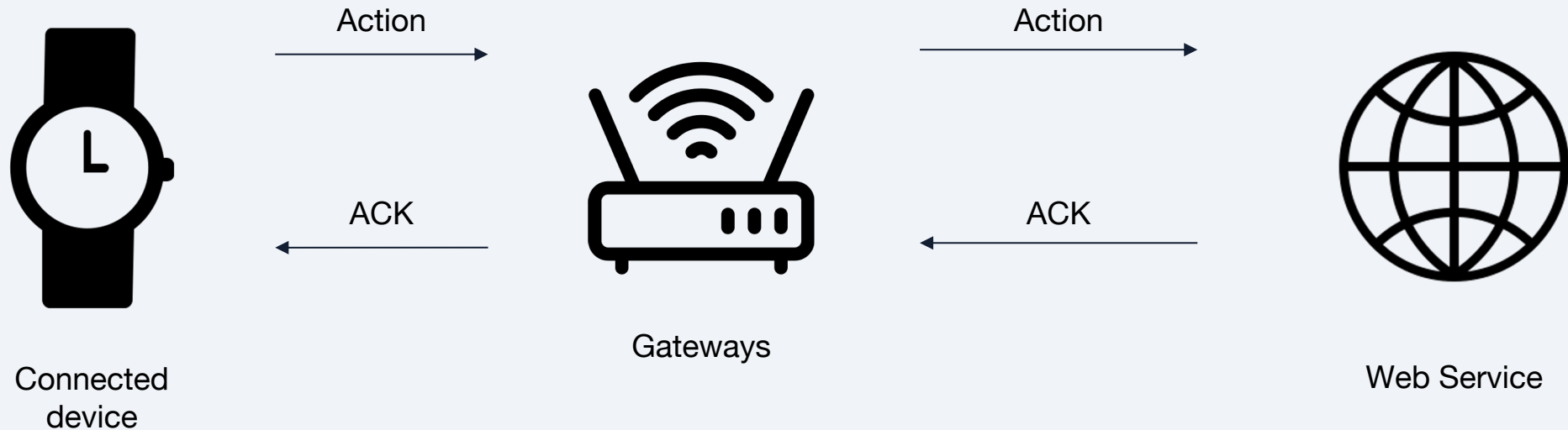




Firmware Team Makeup

We need more than firmware engineers to ship
firmware successfully

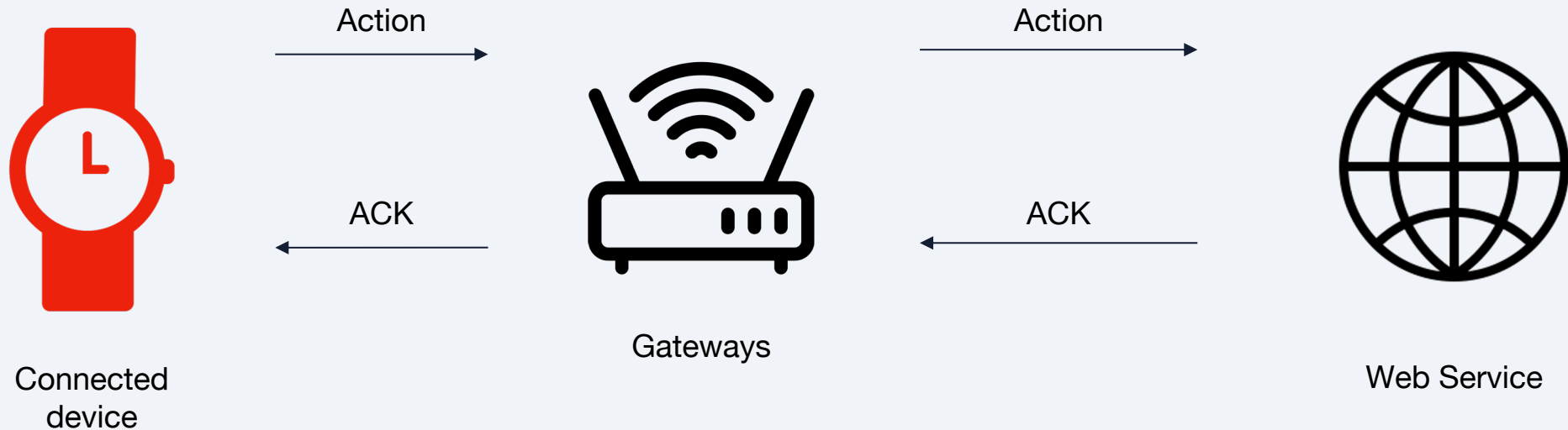
The Hardware Experience Today



**Companies need hardware engineers,
firmware engineers, software engineers,
cloud engineers, and data engineers**

to build the product experience

The Hardware Experience Today



**Companies need hardware engineers,
firmware engineers, software engineers,
cloud engineers, and data engineers**

to build the product experience

***Monitoring the hardware is
almost always an afterthought***

Monitoring the hardware

- Firmware is never perfect
- Customers will have issues
- Security vulnerabilities
- Issues will need to be triaged, reproduced, debugged, and fixed
- Firmware updates

If not prepared for, these activities will consume feature development

**Customer
experiences
issue**



**Contacts
support**



Modern Firmware Team

- **Electrical Engineers**
 - Build & test the hardware
- **Firmware Engineers**
 - Write and debug the firmware
 - Generate debug info, such as logs and core dumps
- **QA & Test Engineers**
 - Find and file issues
- **Software Engineers**
 - Services to accept and store logs, metrics, and core dumps, and query them on demand
- **DevOps Engineers**
 - Make sure the monitoring services stay running
- **Data Engineer**
 - Query the data and find trends

Customer
experiences
issue



Contacts
support



Triaged,
investigated,
fixed, &
deployed

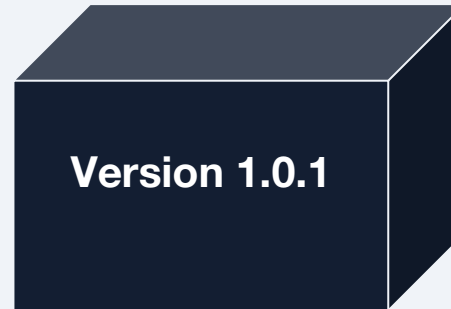
Normal Firmware Experience Getting Data

Firmware team wants to track Wi-Fi RSSI

Firmware Team

```
typedef struct {  
    // Other Metrics  
    ...  
    int16_t rssi;  
} FirmwareMetrics;
```

Add Metric



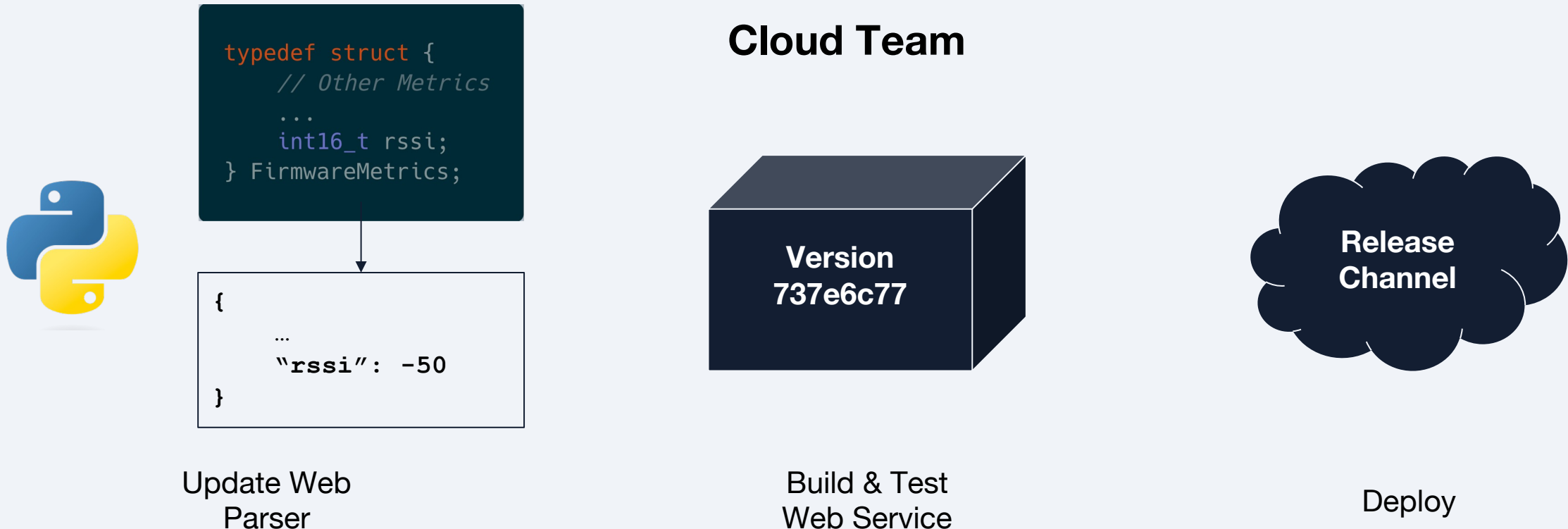
Build & Test
Firmware



Deploy

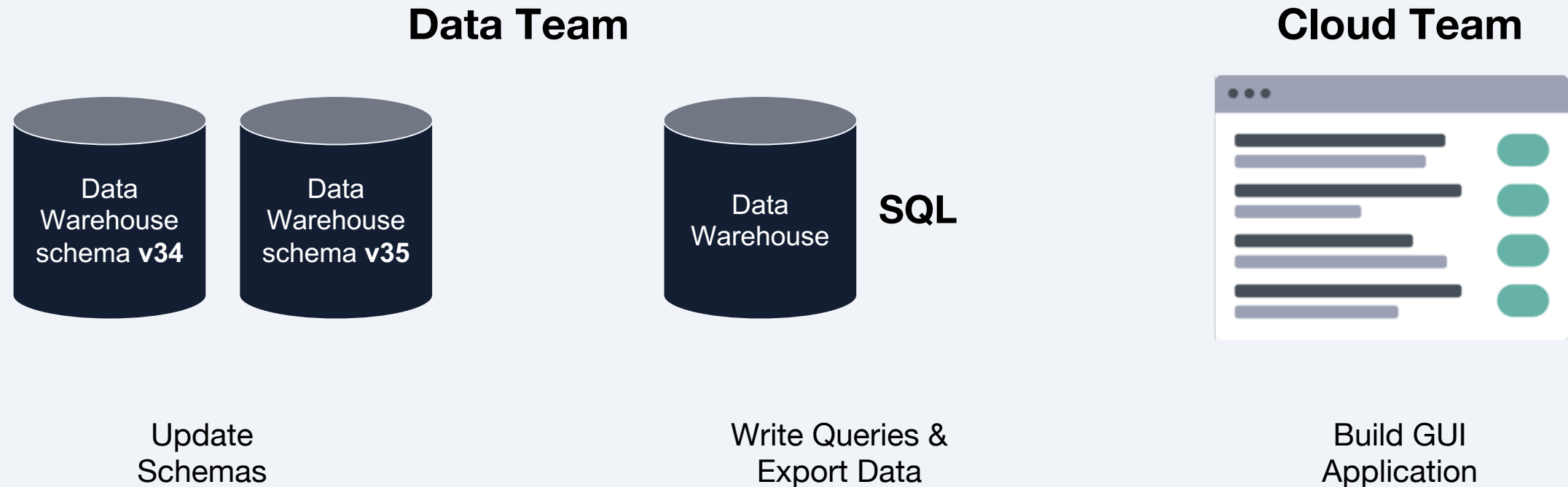
Normal Firmware Experience Getting Data

Firmware team wants to track Wi-Fi RSSI



Normal Firmware Experience Getting Data

Firmware team wants to track Wi-Fi RSSI



Normal Firmware Experience Getting Data

New Firmware with RSSI

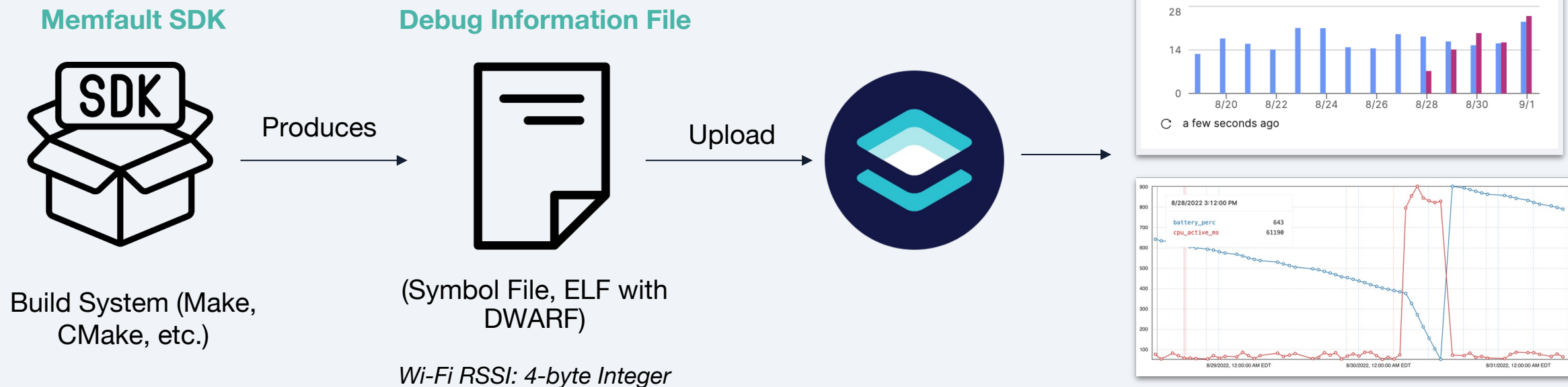
1-12 weeks

Into the Data Warehouse

Extra 1-4 weeks!

New Metrics in Memfault

- Firmware engineers can add metrics with 2 lines of firmware code. No extra engineers needed.
- SDK bundles all information needed into the Debug Information File (ELF)





Debugging in Production

With 10k and more devices in the field, there will be bugs, and they will be weird.

Poll #1

How many
issues have
you seen
surface **only** in
production?

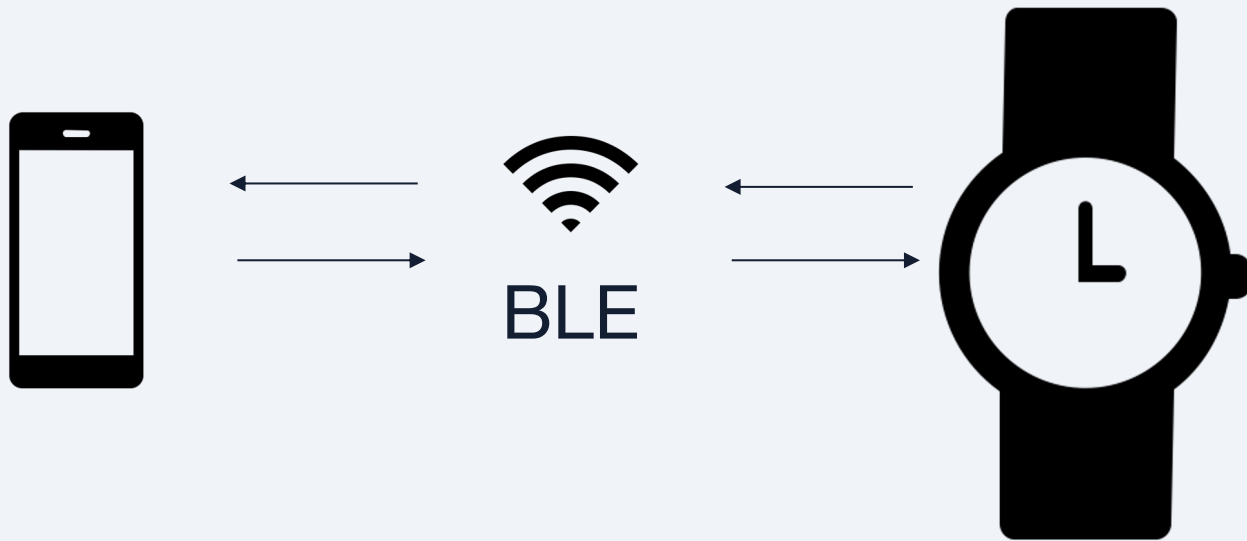
A. 0

B. 1-5

C. 5-10

D. 10 or more

Small & Controlled Environments



It works!

Unbounded Environments



Did it ever work?

Impossible to simulate all environments

- Debugging in production is the only way
- A single company deploys end devices, bridges, gateways, and towers, and is control of all updates of devices
- Do you really control it?
 - Radio interference, weather, power source, hardware quality, user error, and more



1 in 10,000 hour bugs

- Bug occurs once every 10,000 hours
- Takes **416 days** to see it on a single device

*With 10,000 devices that issue is hit **every hour***

*With 1 million devices...**every 36 seconds***

Catastrophic issues might *never* be seen during internal testing



Bugs will always exist

- No firmware is free from bugs
- Vendor libraries and RTOS frameworks will have bugs and security patches
- Firmware updates are a necessary evil

As a first step, track the number of issues
(Crashes, disconnects, asserts, etc.)

Capture as much data as feasible



Webinar: How to Monitor Your IoT Devices at Scale



How to Monitor IoT Devices at Scale

Tyler Hoffman, Co-Founder & Engineer
December 3, 2020

[Watch: How to Monitor IoT Devices at Scale](#)



Scaling Data

Ingesting data is straightforward. Making sense of it is *the hard part*

IoT is a different beast



100 servers

100 metrics
each minute

600k data
points per hour



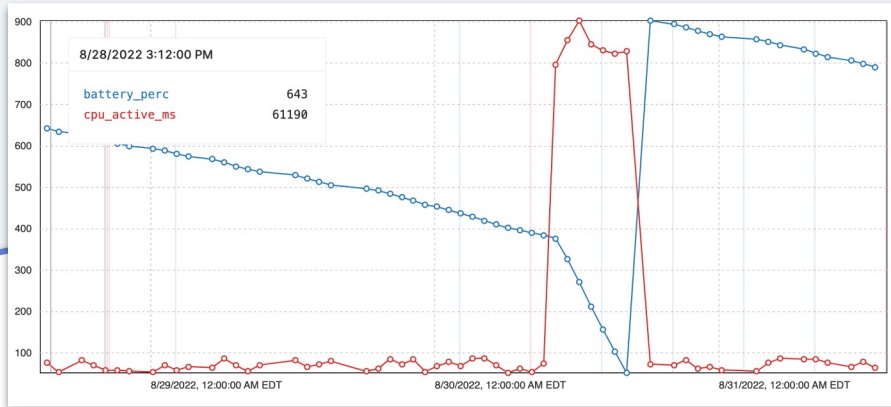
1m devices

100 metrics
each minute

**6 billion data
points per hour**



Data from 1 Million Devices



100 4-byte metrics
100 samples per day
1 million devices
= 40GB per day

30 day retention = 1.2TB

Lvl Message

```
| Initializing Accel Subsystem  
| Launching sleep tracking app  
| Wifi Connected.  
| Analyzing Raw Algo Data (50B)  
| Wifi Unavailable. Retrying in 5 min  
| sleep tracking app closed  
| Launching sleep tracking app  
E Temp Sensor I2C transaction timeout, rv=-8
```

1 64 kB log
1 per day
1 million devices
= 1 million log files at 64GB

30 days retention = 2TB

Data from 1 Million Devices

- AWS S3 doesn't apply here
- Hosted AWS databases - max of 16TB
- Need a time-series database for metrics
- NoSQL database for logs
- S3 for coredumps
- **Downsample data as soon as possible**
- Materialized views, proper indexes, compression, and partitioning of tables will be critical



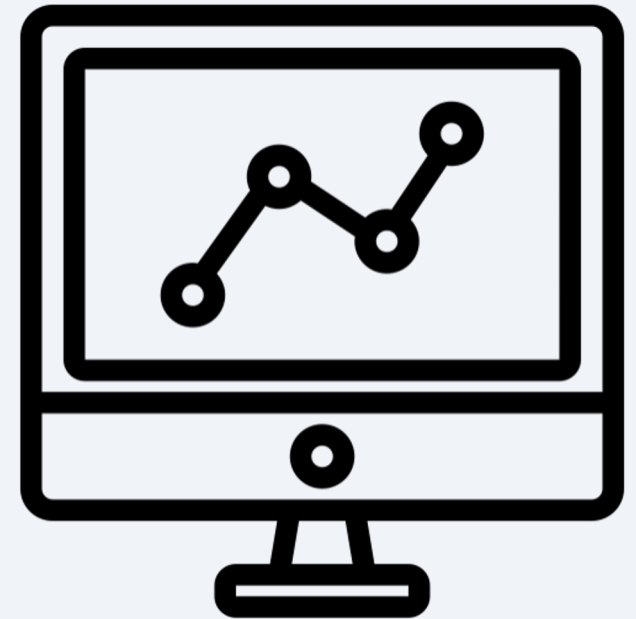
Longer Time Windows for Metrics

Device Hourly Heartbeat

Firmware Version: v1.0.1	MainTaskMs	WifiConnectedMs
Time: 1605910419	1243983	3012321
Seconds Elapsed: 3600	RSSI	...
Device Serial: DA143532	-57	

Try Fleet Sampling

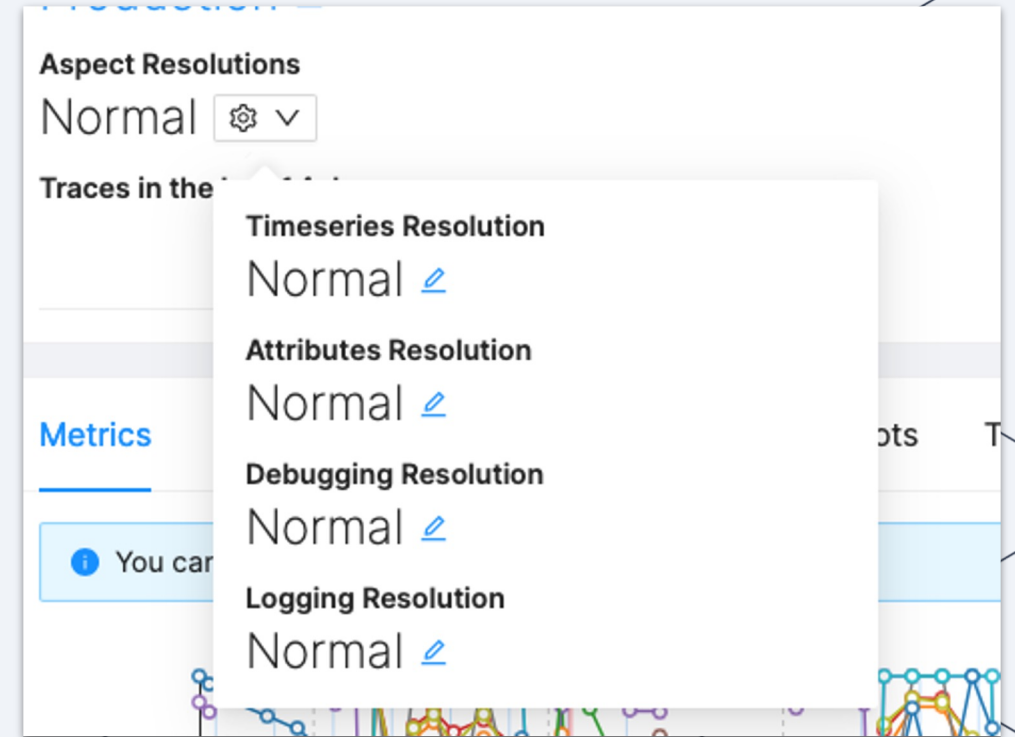
- Instead of actively monitoring 1 million devices...
- Track a subset of them, ~50k, and rotate every day.
- Will see the majority of the issues, but cut down costs dramatically
- Can use different frequencies for different information
 - Uptime metric continuously
 - Metrics sometimes
 - Coredumps & logs infrequently



Fleet Sampling

- 1 millions devices
- Even just 1% reporting in, the 1 in 10,000 bugs happen each hour

If you have a coredump of the issue, you ***only need one occurrence***





Usable Systems

All teams should be able to use the monitoring tools and infrastructure used by the firmware team

Poll #2

**At your company,
how difficult is it
for anyone to dig
into firmware
monitoring data?**

- A. We don't collect monitoring data**
- B. It's challenging for everyone**
- C. Only the firmware team can do
this easily**
- D. Very easy for entire company**

**Once the data pipe is there,
every team will want to use it**

Tools for Everyone!

Engineering

- Root cause of bugs
- No need to reproduce
- Prioritize issues
- Correlate regressions to software versions
- Dig deeper into power and performance issues

Support

- View history of device's issues, metrics, and logs
- Proactively reach out to customers
- Not bother engineers with same issues

Product & Exec

- Determine success or failure of a firmware update
- Understand prevalence of issues
- Battery life & connectivity issues ruin customer experiences

Local environments don't scale



Make sure to have SSH keys

```
$ git clone git@github.com:acme-inc/awesome-firmware.git
```

```
$ cd awesome-firmware/
```

!! DON'T RUN SUDO !! Don't use your system Python!

```
$ pip install virtualenv
```

```
$ virtualenv venv
```


Everyone forgets this

```
$ source venv/bin/activate
```

Is everything ever properly listed here?

```
$ pip install -r requirements.txt
```

```
$ python tools/scripts/debug/do-something --arg --arg --arg
```

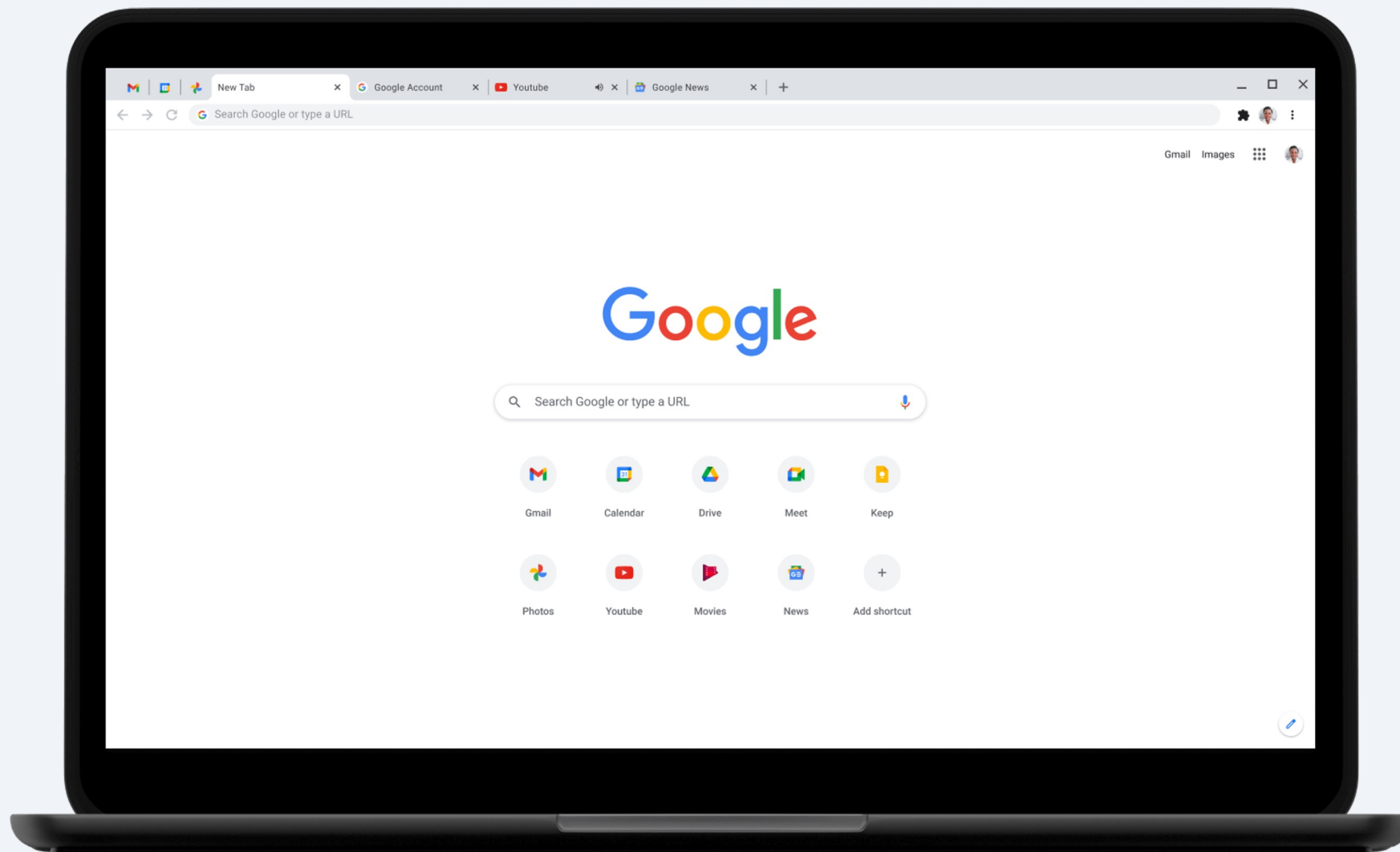



Not everyone's a developer

- Can't give everyone at the company GitHub access.
- Not everyone uses a terminal
- Environments are difficult to install and maintain
- Toolchains are difficult to find and distribute (especially Windows)
- Docker doesn't solve all the problems here

The background is a solid teal color. It features a pattern of thin, white, overlapping geometric shapes. These shapes include diamonds, rounded rectangles, and wavy lines, creating a modern, abstract design. The text is centered in the middle of the image.

**Everyone should use the
one approved way**



We turned to web applications

- Login with SSO (Google, Okta, LDAP)
- No local setup necessary
- One engineer can empower the company
- Python locally, Python in webapps
- Simplest applications can be 10-50 lines of code



Web applications sound great...

- FW engineers were managing our web applications, infrastructure, databases, security, and more
- Still responsible for firmware tasks
- Needed a SW or DevOps engineer on our team



Start Simple

- Get access to deploy behind VPN
- Use Python or Ruby and HTML
- Doesn't have to be pretty
- Script-like applications are great!

Web application

```
python tools/decode_logs.py  
-log-file <uploaded_file>
```

Firmware Version:

Paste register addresses here:

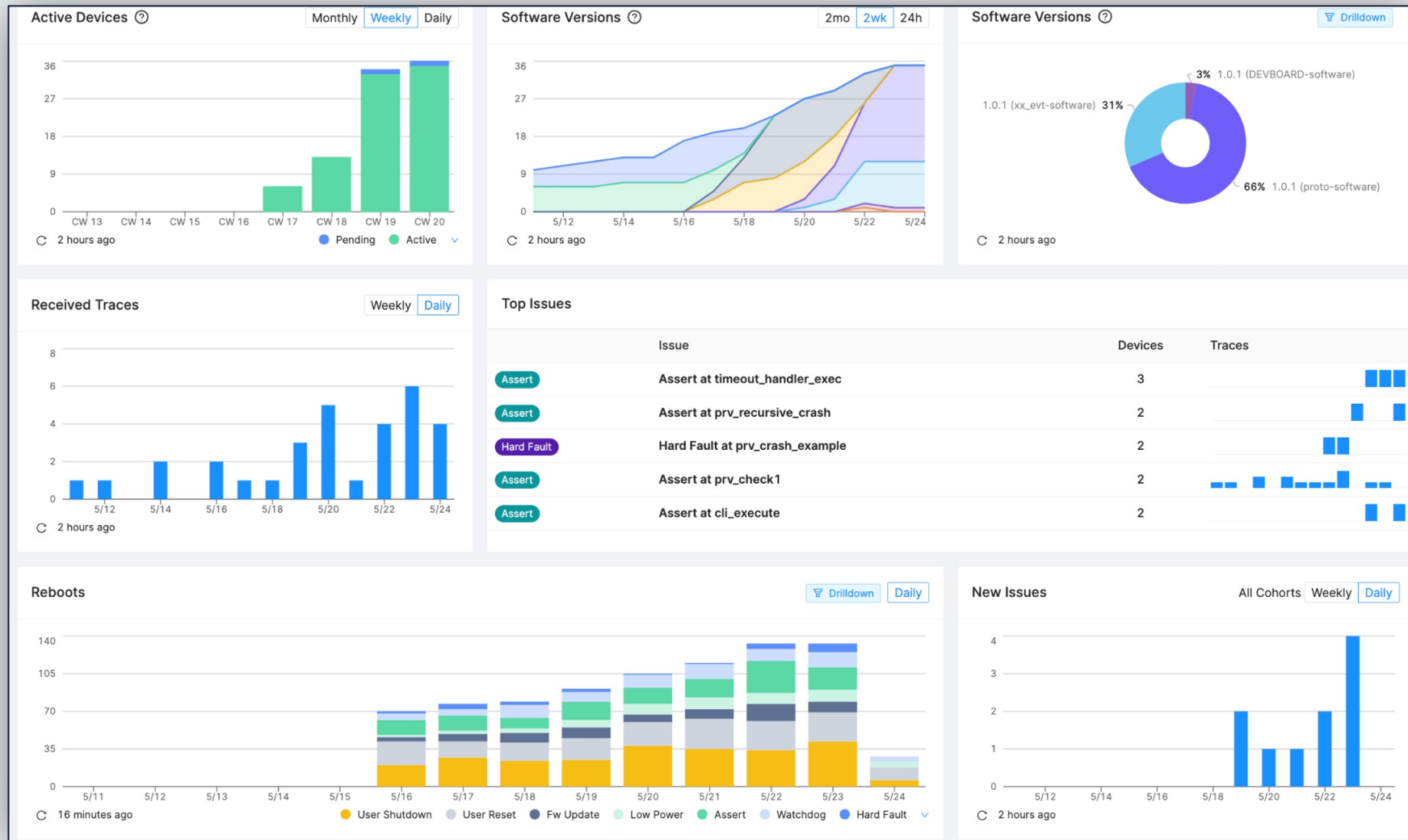
```
0x12345678,  
0x523711,  
0x80002145,
```

Symbolize

```
flash_write_bytes,  
prv_write_to_filesystem,  
main,
```

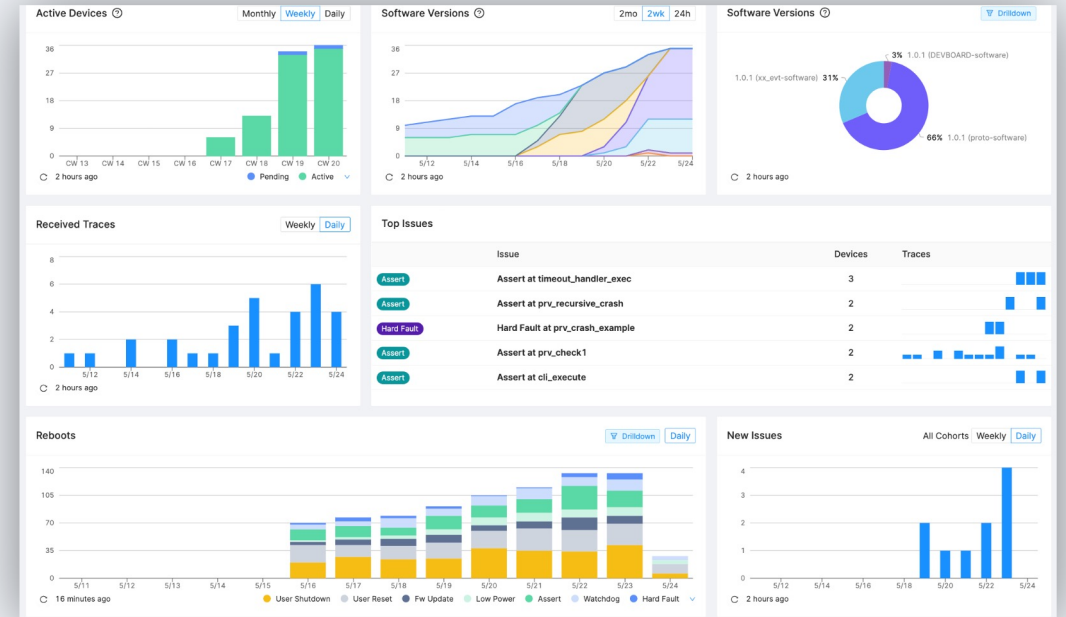


Memfault



Memfault for Everyone

- Memfault is a suite of tools, primarily a web application
- Stable API for integrations with CI and other tools
- Usable by engineers, support staff, and product teams



Four Topics Covered Today

**Firmware
Team**



**Debugging in
Production**



Scaling Data



**Usable
Systems**



Q&A

Would appreciate filling out the [survey](#) at the end.

*It will appear in browser when the webinar is over and
we will have it in the follow-up email.*



Memfault