

PARTNER
WEBINAR



NORDICTECH
WEBINARS

Remotely debug, monitor,
and update Nordic-powered
IoT devices with Memfault

02/56 → OPERATING MODE

01/55

Today's hosts

Heiko Behrens



Head of Product



Ali Aljaani

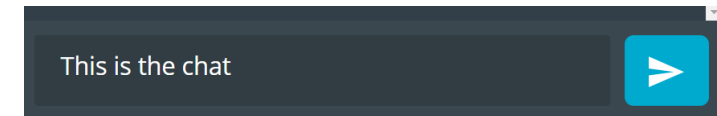
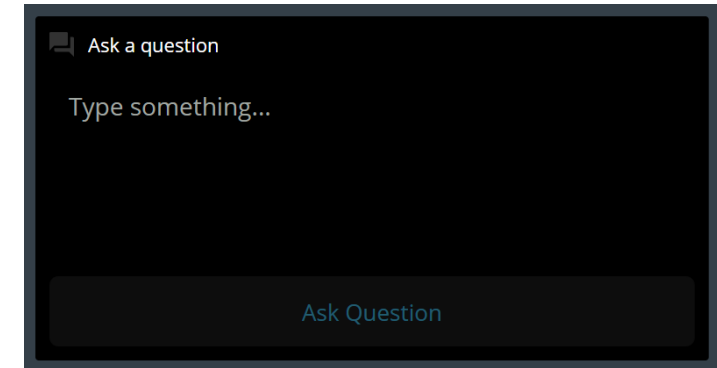


Product Marketing
Engineer



Practicalities

- Duration: 50-60 mins
- Questions are encouraged!
- Please type questions in the top of the right sidebar
 - All questions are anonymous
 - Try to keep them relevant to the topic
- We will answer questions towards the end
- The chat is not anonymous, and should **not** be used for questions
- If you have more questions:
 - Go to DevZone for Nordic related questions
 - Go to <https://memfault.com/contact/> for help with Memfault platform
 - A recording of the webinar will be available together with the presentation at webinars.nordicsemi.com



{ DevZone

Today's Plan



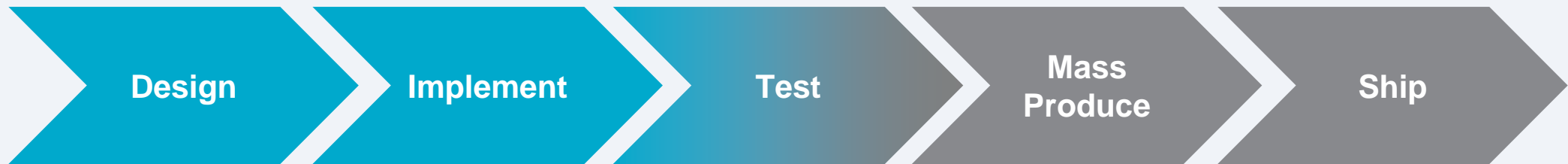
What is Memfault?

Live Integration Demo

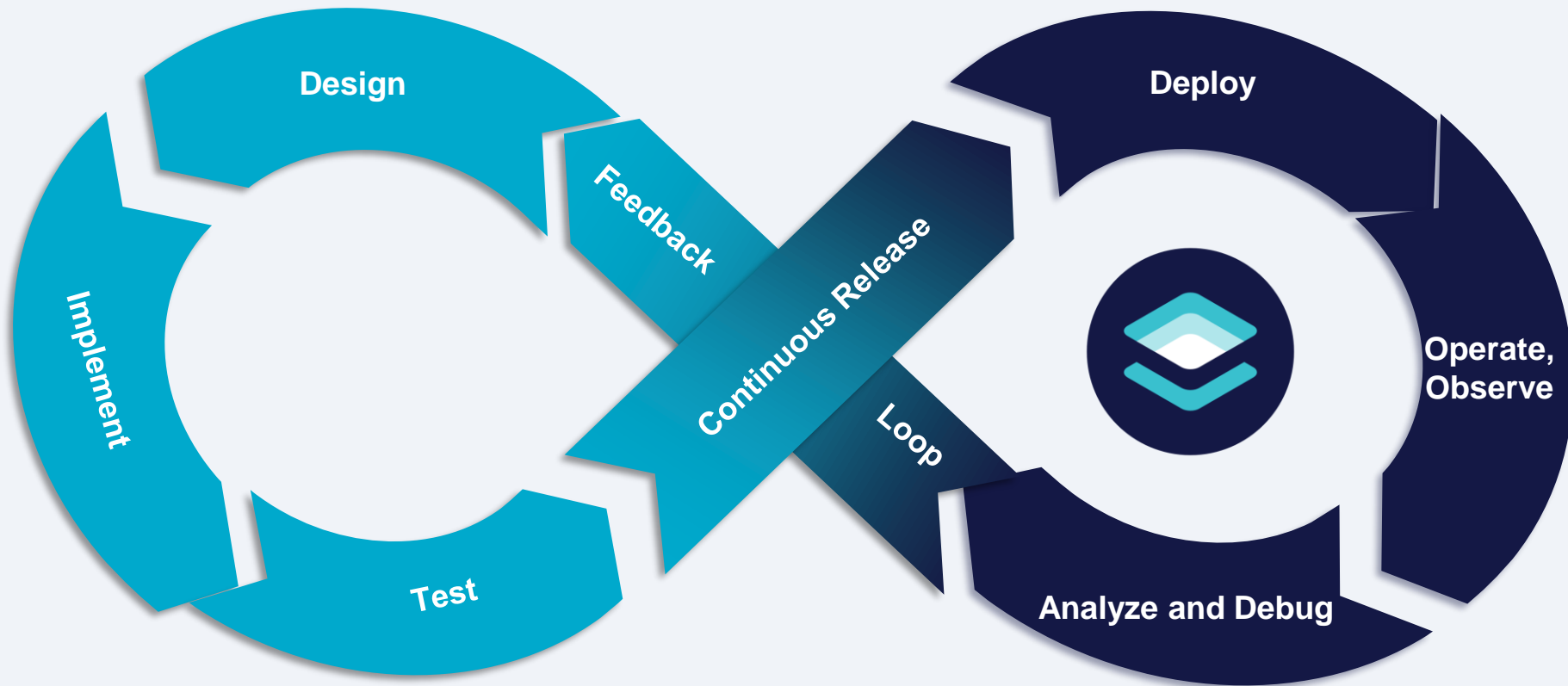
Technical Deep Dive

**Memfault is the first
observability platform for
connected products**

Hardware Development Process



A Better Way



Better Firmware with Memfault in Three Steps



Continuously monitor devices

View device and fleet-level metrics like battery health and connectivity in real-time dashboards.



Remotely debug firmware issues

Resolve issues more efficiently with automatic detection, alerts, deduplication, and actionable insights sent via the cloud.



Surgically deploy OTA updates

Once bugs are fixed, deploy updates directly to customer devices in the field for quick issue resolution.

**Customers reduce the
number of resets per
device by 90%**

Memfault is included **for free
in the nRF Connect SDK v1.6**



**Let's implement Memfault on
the nRF9160!**

Remote Debugging

Automatically capture errors across the entire OS and all apps.

- ◇ Hardfault
- ◇ Watchdog Hang
- ◇ Stack Overflows
- ◇ SDK Errors
- ◇ Software Asserts
- ◇ Memory Faults
- ◇ Bus Faults

State

Logs

Threads

accel-workq (2) STACK OVERFLOW RUNNING

0 compute_fft in .../src/fft.c at line 10

1 sleep_algo_compute_sleep_time in .../src/sleep_algo.c at line 12

2 process_accel_data_worker_task in .../src/accel_data.c at line 106

3 z_work_q_main in .../zephyr/lib/os/work_q.c at line 32

4 z_thread_entry in .../lib/os/thread_entry.c at line 29

5 0xaaaaaaaa

Thread 3 SUSPENDED

idle (4) READY

logging (5) SUSPENDED

net_mgmt (6) BLOCKED

rx_workq (7) BLOCKED

shell_uart (8) BLOCKED

sysworkq (9) BLOCKED

tx_workq (10) BLOCKED

Registers & Locals

Globals

A dft_out = 0x2000a900 <my_sta

L i = 400

A num_samples = 536912536

A raw_samples = 0x3128115f

L tmp = {1, 222, 7, 84}

R \$r0 = long 536912536 (0x200

R \$r1 = long 1372324912 (0x51

R \$r2 = long 1372324919 (0x51

R \$r3 = long 536912832 (0x200

R \$r4 = long 536912508 (0x200

R \$r5 = long 536914136 (0x200

R \$r6 = long 0 (0x00000000)

R \$r7 = long 536912488 (0x200

R \$r8 = long 0 (0x00000000)

R \$r9 = long 0 (0x00000000)

R \$r10 = long 0 (0x00000000)

R \$r11 = long 0 (0x00000000)

Detailed Error Reporting

```
/* header file */
MEMFAULT_TRACE_REASON_DEFINE(custom_error_1);

/* C file */
void ble_le_process_ll_pkt(...) {
    // ...
    if (invalid_msg_id) {

MEMFAULT_TRACE_EVENT(custom_error_1);
    // ...
    }
    // ..
}
```

Add new error types quickly with **2 lines of code.**

- ◇ Hardfault
- ◇ Watchdog Hang
- ◇ Stack Overflows
- ◇ SDK Errors
- ◇ Software Asserts
- ◇ Application Errors

Device Monitoring

- Pre-populated with 15 metrics core to debugging on Nordic devices.
- Instant access to the exact device-level data you need to resolve bugs faster.

AllTasksCpuUsage
at_cmd_unused_stack
connection_poll_unused_stack
lte_connection_loss_count
lte_time_to_connect
MemfaultSdkMetric_IntervalMs
MemfaultSdkMetric_UnexpectedRebootCount
Ncs_AtCmdUnusedStack
Ncs_ConnectionPollUnusedStack
Ncs_LteConnectionLossCount
Ncs_LteTimeToConnect
switch_1_toggle_count
Switch1ToggleCount
TimerTaskCpuUsage
TimerTaskFreeStack



Add new metrics with 2 lines of code

```
/* header file */
MEMFAULT_METRICS_KEY_DEFINE (BtBytesSent,
    kMemfaultMetricType_Unsigned);

/* C file */
void bluetooth_driver_send_bytes(const void
                                *data, size_t
                                data_len)
{
    memfault_metrics_heartbeat_add(
        MEMFAULT_METRICS_KEY(BtBytesSent),
        Data_len
    );
    // [ ... code to send bluetooth data ... ]
}
```

◇ Battery level

◇ Free memory

◇ Bluetooth LE
Statistics

◇ Flash Statistics

◇ CPU Statistics

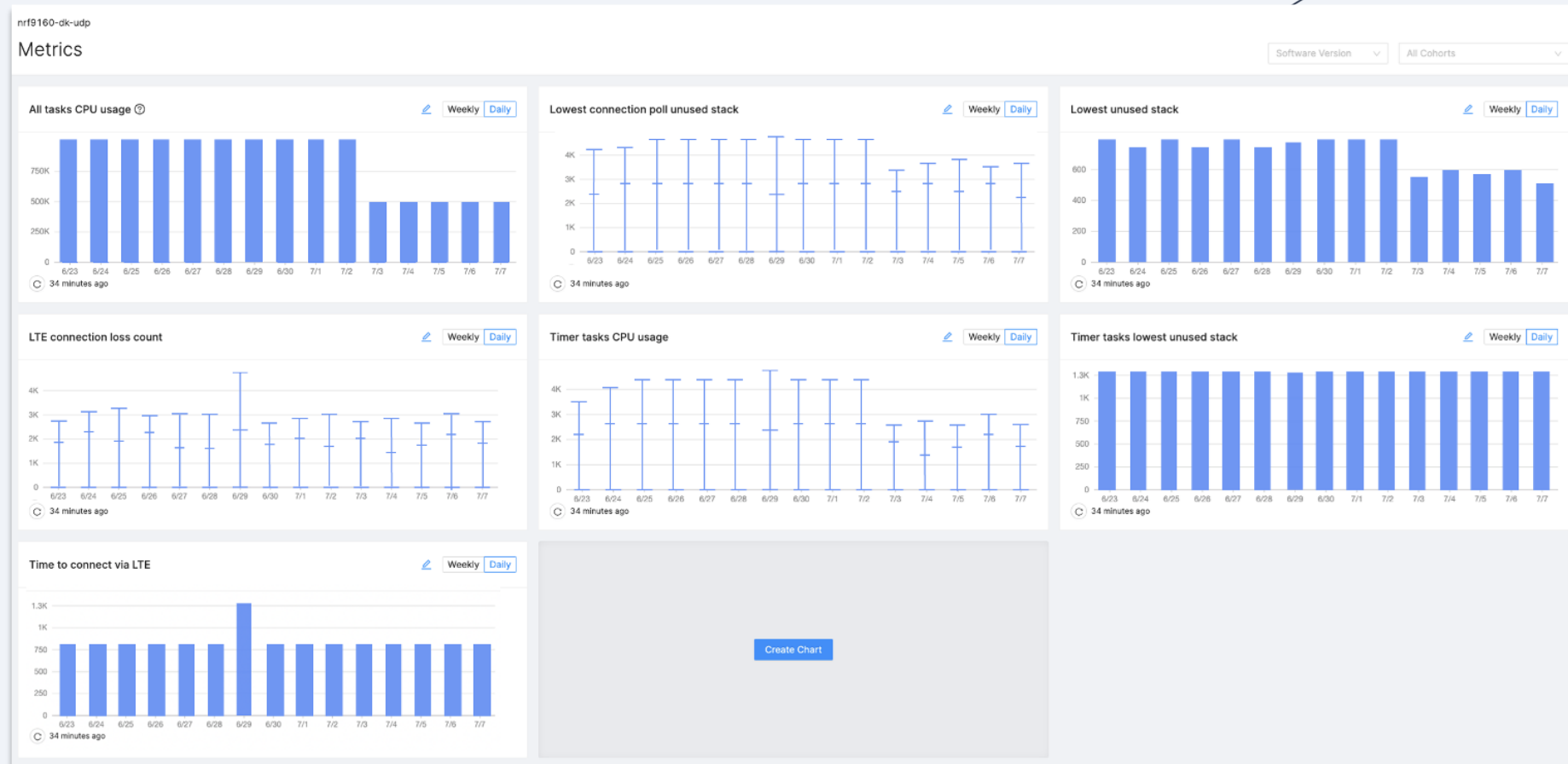
◇ RTOS Statistics

Continuous Fleet Monitoring

Nordic customers have preset & editable fleet-wide data on the metrics they care about.

All Tasks CPU Usage

- Lowest connection poll unused stack
- Lowest unused stack





How does it work?

Life of a coredump

1 Device experiences an issue

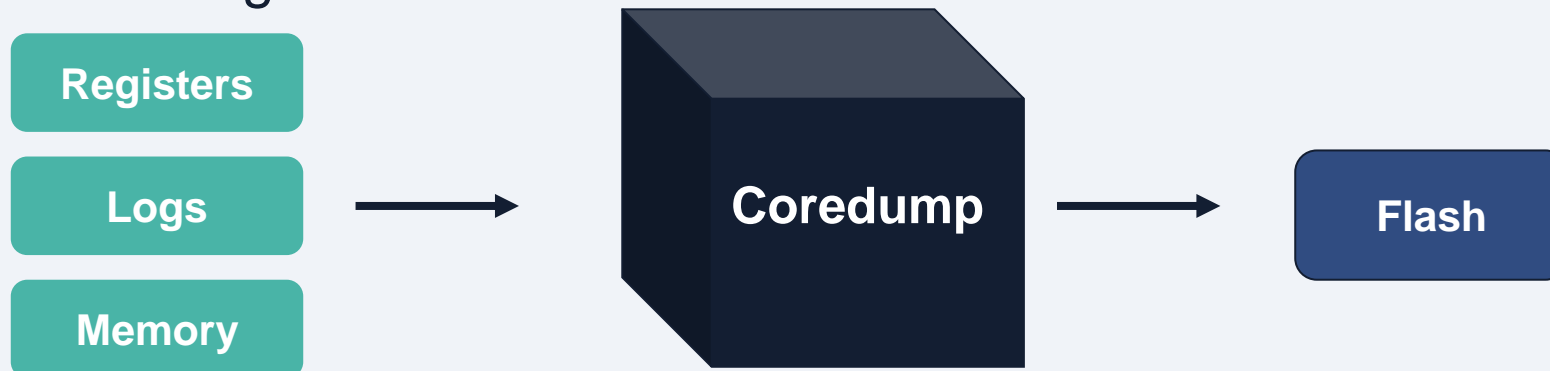
```
void foo(void) {  
    void (*a_ptr)() = NULL;  
    a_ptr();  
}
```



2 The Memfault SDK is invoked

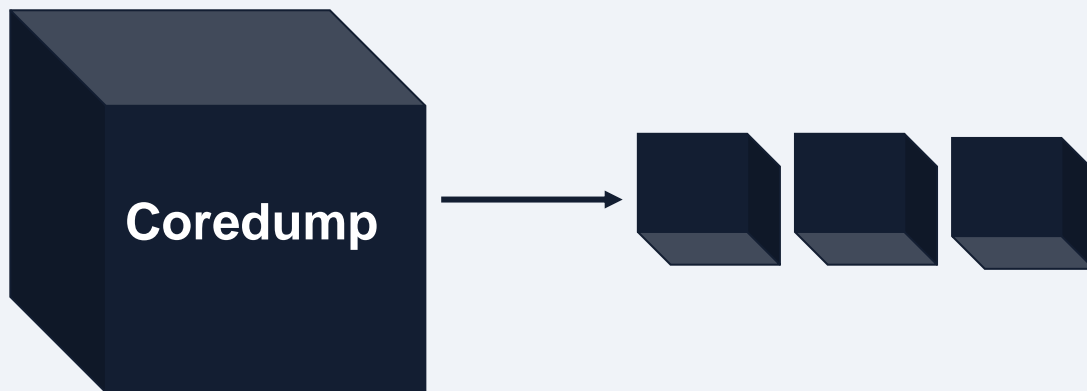
```
void Hardfault_Handler() {  
    ...  
    memfault_fault_handler(regs, reason);  
}
```

3 Memfault saves useful data to non-volatile storage

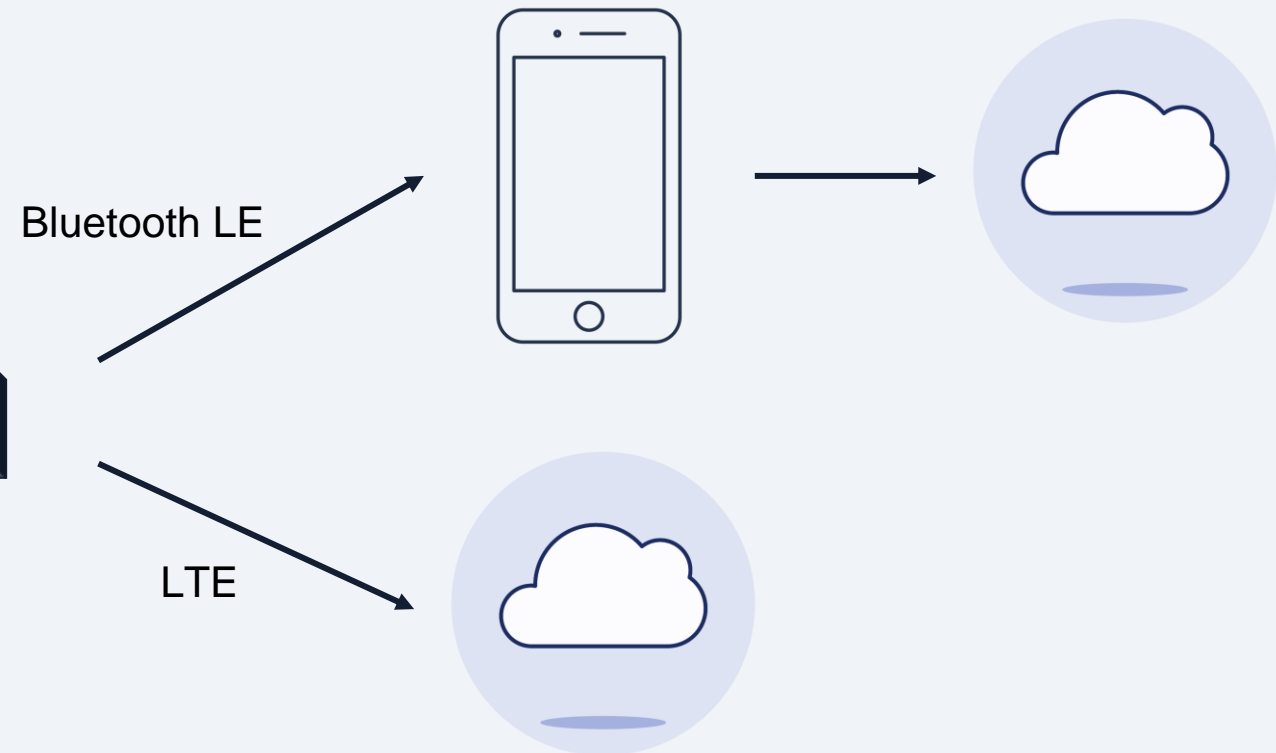


Life of a coredump (cont'd)

4 The data is broken into chunks

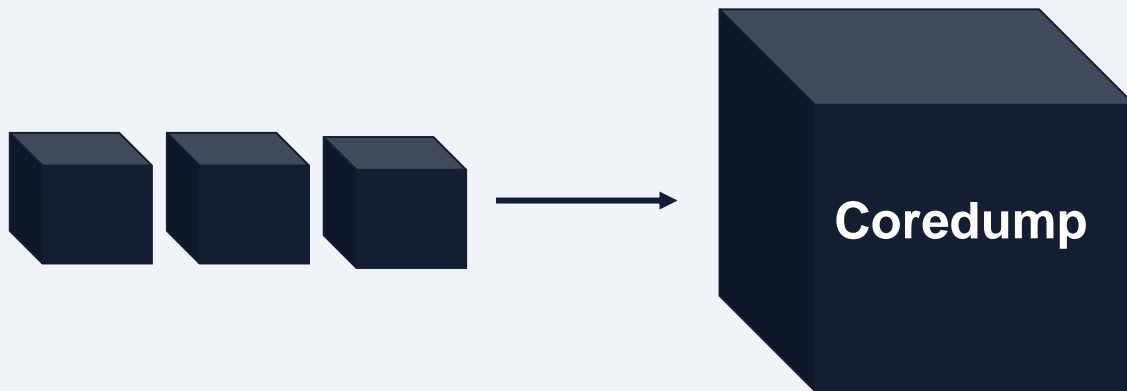


5 The chunks are sent to the Memfault cloud

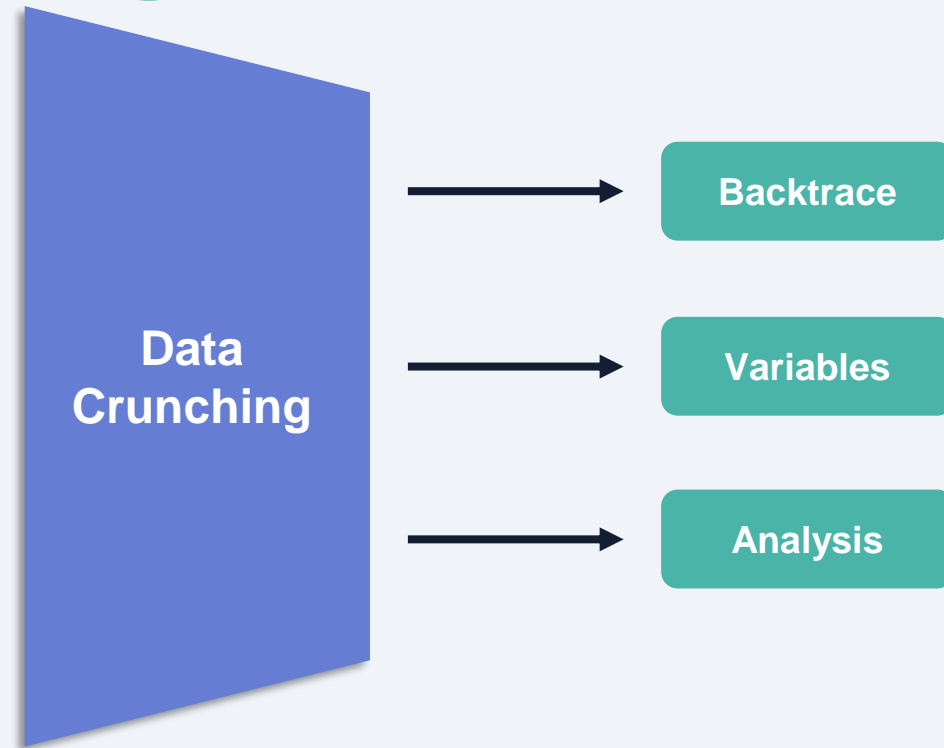


Life of a coredump (cont'd)

⑥ Memfault reassembles the data



⑦ Memfault then analyzes, symbolicates, and deduplicates it



Where do I start?

- Sign up at <https://memfault.com/register/nrf>
- Memfault docs: <https://docs.memfault.com/docs/mcu/nrf-connect-sdk-guide>
- Nordic docs:
https://developer.nordicsemi.com/nRF_Connect_SDK/doc/latest/nrf/include/memfault_ncs.html
- Memfault SDK: <https://github.com/memfault/memfault-firmware-sdk>
- Complete Sample for Thingy:91 and nRF9160 DK:
https://developer.nordicsemi.com/nRF_Connect_SDK/doc/latest/nrf/samples/nrf9160/memfault/README.html
- Contact us at support@memfault.com
- Join the Interrupt Slack: <https://interrupt-slack.herokuapp.com/>

Q&A

Register for upcoming Nordic Tech Webinars

www.nordicsemi.com/webinars