Memfault

Getting Started with Memfault for MCU

François Baldassari, CEO, Memfault

Agenda

◇ **Brief Memfault**
Overview

◇ **Live Integration**
Demo

◇ **Q & A**

Remote Debugging

◇ Catch issues sooner

Detect issues before your customers with automatic data extraction, aggregation and real-time alerts.

◇ Resolve issues faster

Find a fix to the most critical issues with deduplication and grouping and a snapshot of all hard data.

◇ Make products stronger

Find a fix to the most critical issues with deduplication and grouping and a snapshot of all hard data.

Continuous Monitoring

◇ **Customize for your needs**

Add custom metrics with just 2 lines of code and configure alerts to track what's most important to you.

◇ **Track release adoption**

Understand the adoption rate for a critical software update with version reporting.

◇ **Scale up or scale down**

Decide how many metrics to collect based on your device's connectivity, power, and bandwidth needs. We handle it all.

OTA **Updates**

◇ **Deploy cohort-based and staged rollouts**

Split devices into specific groups to test updates with beta users, customers facing a bug, or another group before you deploy to all devices.

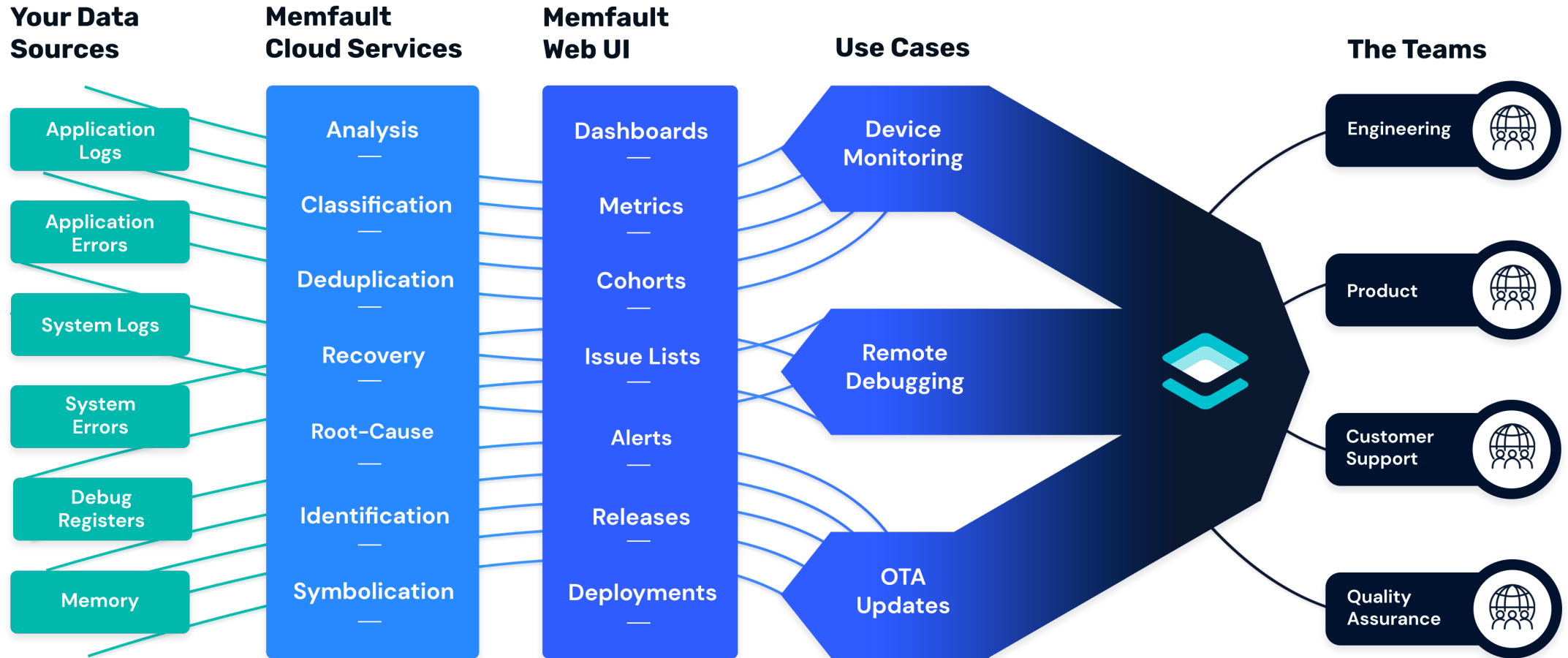◇ **Schedule updates when you want**

Whether you want to send an update while your users are sleeping or first thing in the morning, you can pick specific times for each update.

◇ **Stop faulty updates with one click**

Get notifications about issues in your updates, and rollback the update with one-click before 100% of your fleet is impacted.

How it works

Your Data Sources

- Application Logs
- Application Errors
- System Logs
- System Errors
- Debug Registers
- Memory

Memfault Cloud Services

- Analysis
- Classification
- Deduplication
- Recovery
- Root-Cause
- Identification
- Symbolication

Memfault Web UI

- Dashboards
- Metrics
- Cohorts
- Issue Lists
- Alerts
- Releases
- Deployments

Use Cases

- Device Monitoring
- Remote Debugging
- OTA Updates

The Teams

- Engineering
- Product
- Customer Support
- Quality Assurance

Error Types

Add custom error types with **2 lines of code:**

```c
/* 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);
    // ...
  }
  // ..
}
```

◇ **Hardfault**

◇ **Watchdog Hang**

◇ **Stack Overflows**

◇ **SDK Errors**

◇ **Software Asserts**

◇ **Application Errors**

Metric Types

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**

◇ **BLE Statistics**

◇ **Flash Statistics**

◇ **CPU Statistics**

◇ **RTOS Statistics**

Our SDK

SDK Facts & Figures	
Language	C-99
License	Source Available, BSD
ROM Usage	3 kB
RAM Usage	1.5 kB
Supported Platforms	ARM, Xtensa (more available on request)
Gateway SDK Platforms	iOS, Android, Linux, PC, Mac