



flight



Matt Massicotte
Engineer | @mattie



iOS Crash Reporting

what is a **crash**?

```
// Dereference null
```

```
int *x = NULL;
```

```
*x = 42;
```

```
// Self-termination
```

```
fatalError("ka-boom")
```

```
// Raise Exception
```

```
NSArray *array = @[];
```

```
NSLog(@"%@", array[0]);
```

how are crashes **detected**?

```
// NSUncaughtExceptionHandler
NSSetUncaughtExceptionHandler(myHandler);

// POSIX Signals
sigaction(SIGSEGV, &newAction, &previousAction);

// Mach Exceptions
task_swap_exception_ports(mach_task_self(), ...);
```

crash reports tell you **what**
happened, not **why**

the details of the machine
matter

Stack Unwinding Options

Frame Pointers

simple
common on all architectures
cannot know if it is available

DWARF CFI

highly complex
all possible register mutations
compression / stack-based VM

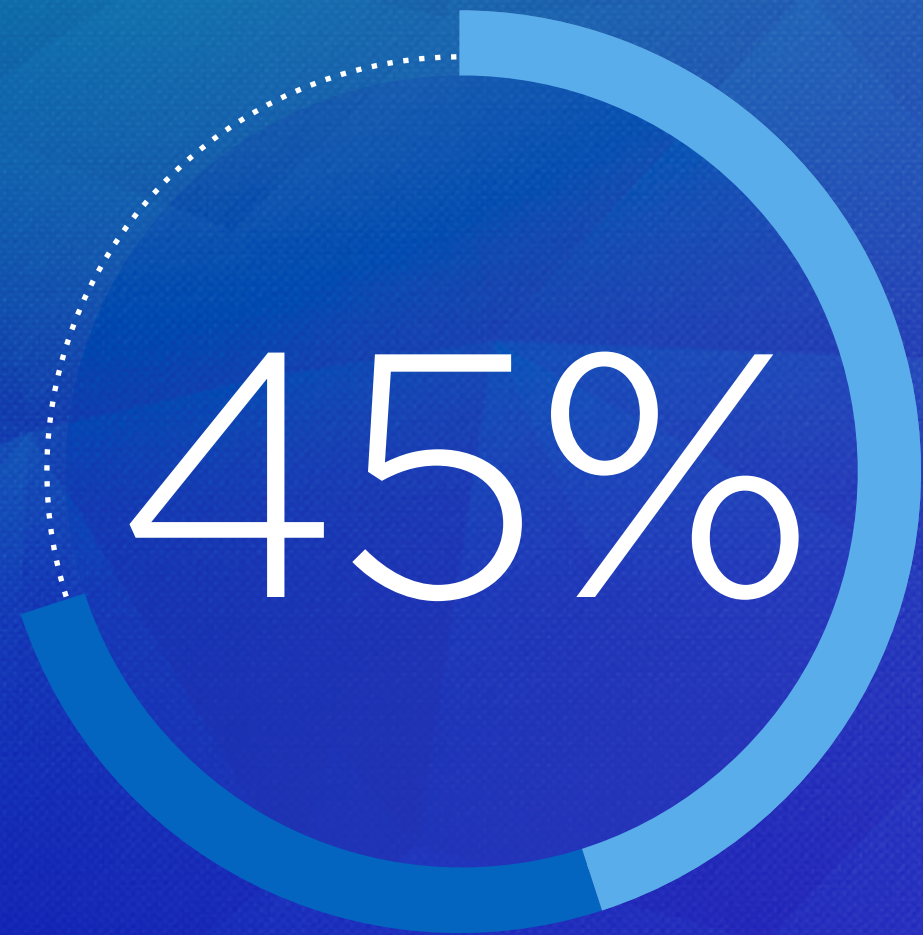
Compact Unwind

manageable complexity
some register mutations
~ 100% of arm64 functions

is in-process crash
reporting **safe**?

nope

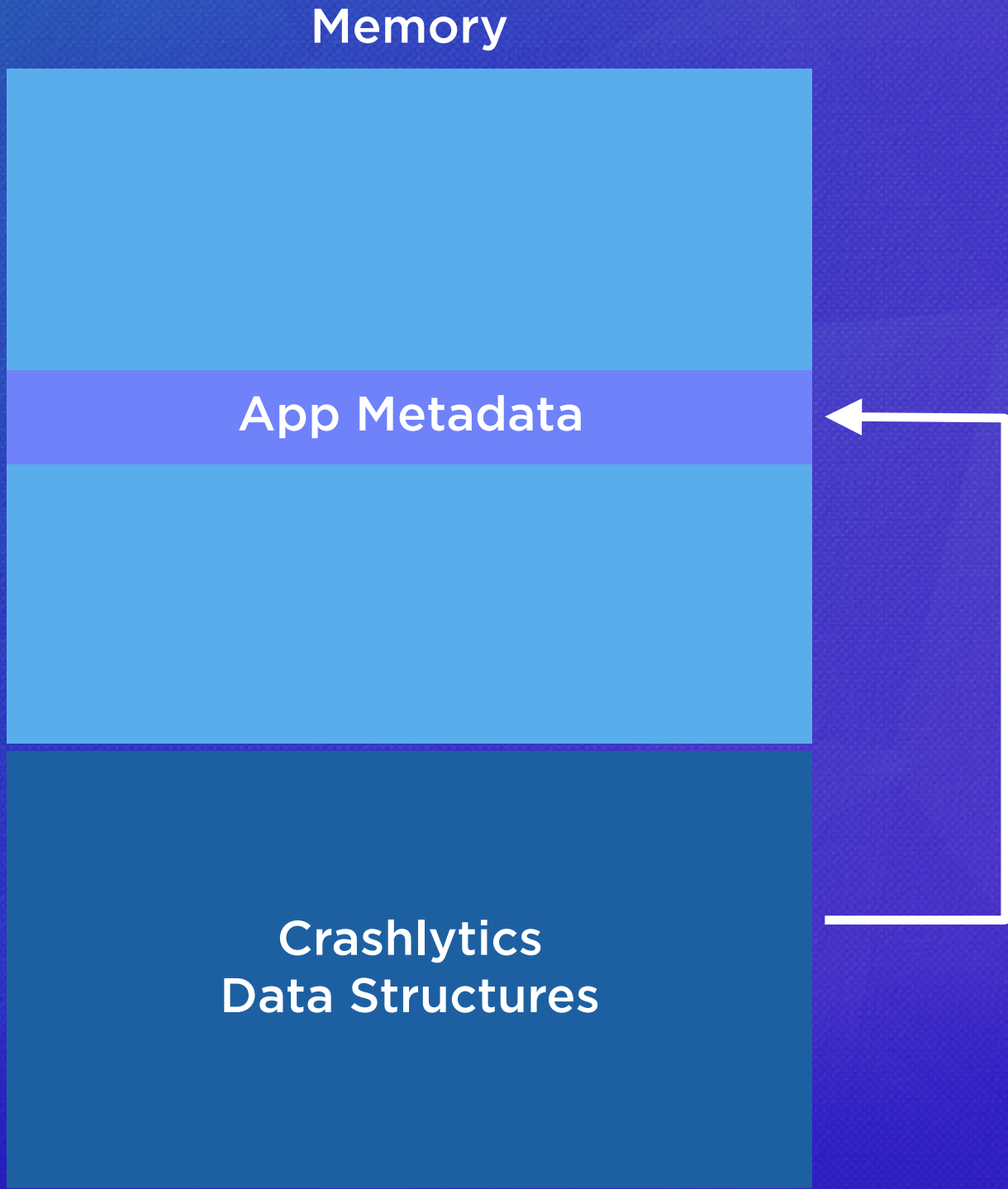
threads are in **arbitrary**
states



of crashes are caused by
invalid memory accesses

the real challenge is
memory corruption

memory corruption
defense strategies



capturing data is just
the **first** step

reports must be
transmitted **reliably**

Crash-on-Launch Example



running in your app is a
big responsibility



Thank You

@mattie

```
config = [NSURLSessionConfiguration backgroundSessionConfiguration:@"..."];  
config.sessionSendsLaunchEvents = NO;
```