



Intro

by: Siarhei Misko

Observability

by: Siarhei Misko

Observability in iOS

by: Siarhei Misko

Logging

by: Siarhei Misko

Optimize!

by: Siarhei Misko

BUILDING OBSERVABILITY IN IOS APPLICATIONS



Aa



Intro

by: Siarhei Misko

Observability

by: Siarhei Misko

Observability in iOS

by: Siarhei Misko

Logging

by: Siarhei Misko

Optimize!

by: Siarhei Misko

Siarhei Misko

iOS Team Lead

siarheimisko@gmail.com





Aa



Intro

by: Siarhei Misko

Observability

by: Siarhei Misko

Observability in iOS

by: Siarhei Misko

Logging

by: Siarhei Misko

Optimize!

by: Siarhei Misko

App stability

Test cases

Test environment

Specific devices



Intro

by: Siarhei Misko

Observability

by: Siarhei Misko

Observability in iOS

by: Siarhei Misko

Logging

by: Siarhei Misko

Optimize!

by: Siarhei Misko

App stability

Test cases

Test environment

Specific devices

Different locations
All network conditions
RTL locale

All supported iOS devices
All supported iPads
All supported macs with Apple Silicon

Different app settings



Aa   



Intro

by: Siarhei Misko

Observability

by: Siarhei Misko

Observability in iOS

by: Siarhei Misko

Logging

by: Siarhei Misko

Optimize!

by: Siarhei Misko

“
**Observability - the
ability of a system to provide
information about its internal
state**”

Intro

by: Siarhei Misko

Observability

by: Siarhei Misko

Observability in iOS

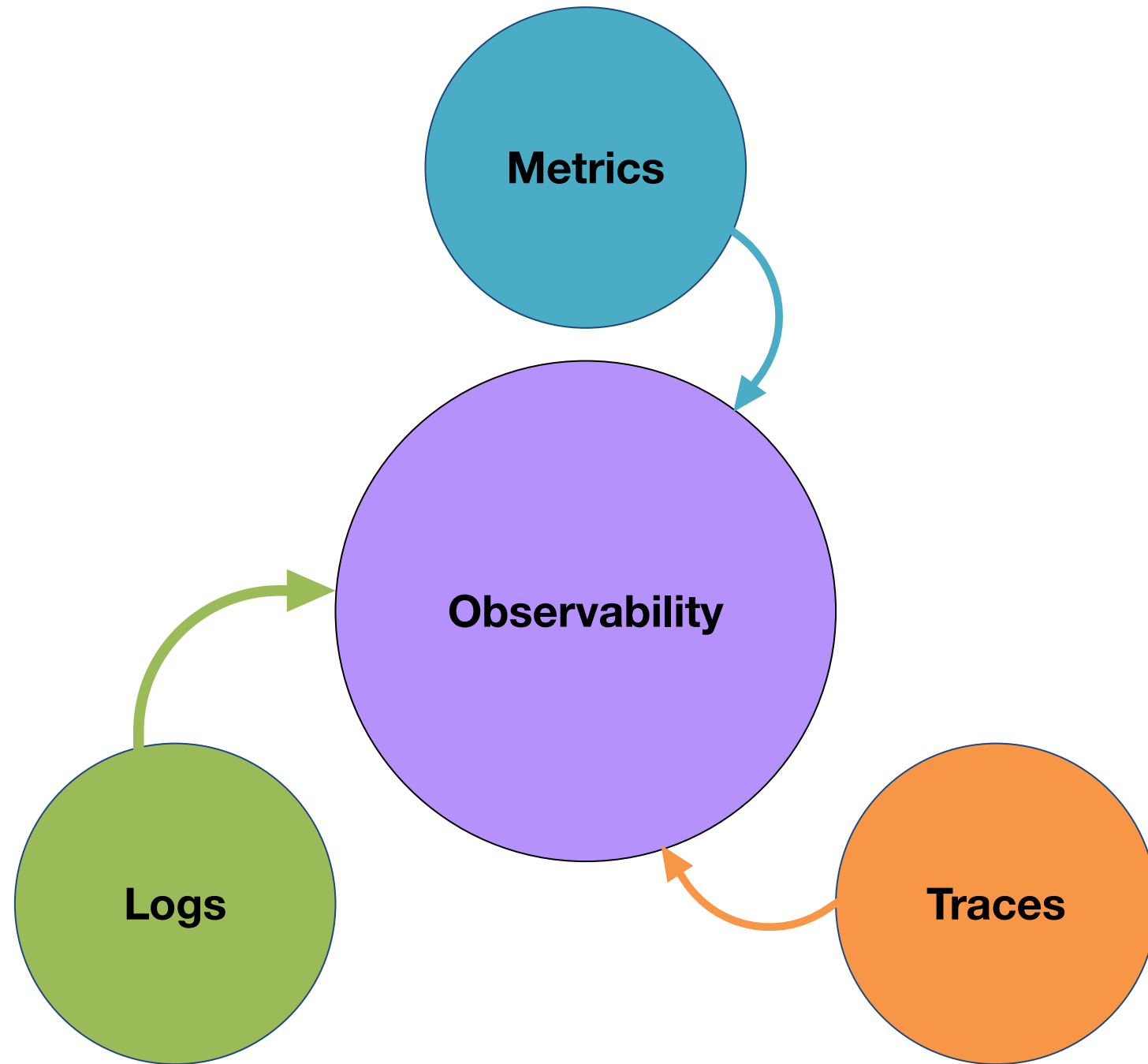
by: Siarhei Misko

Logging

by: Siarhei Misko

Optimize!

by: Siarhei Misko





Intro

by: Siarhei Misko

Observability

by: Siarhei Misko

Observability in iOS

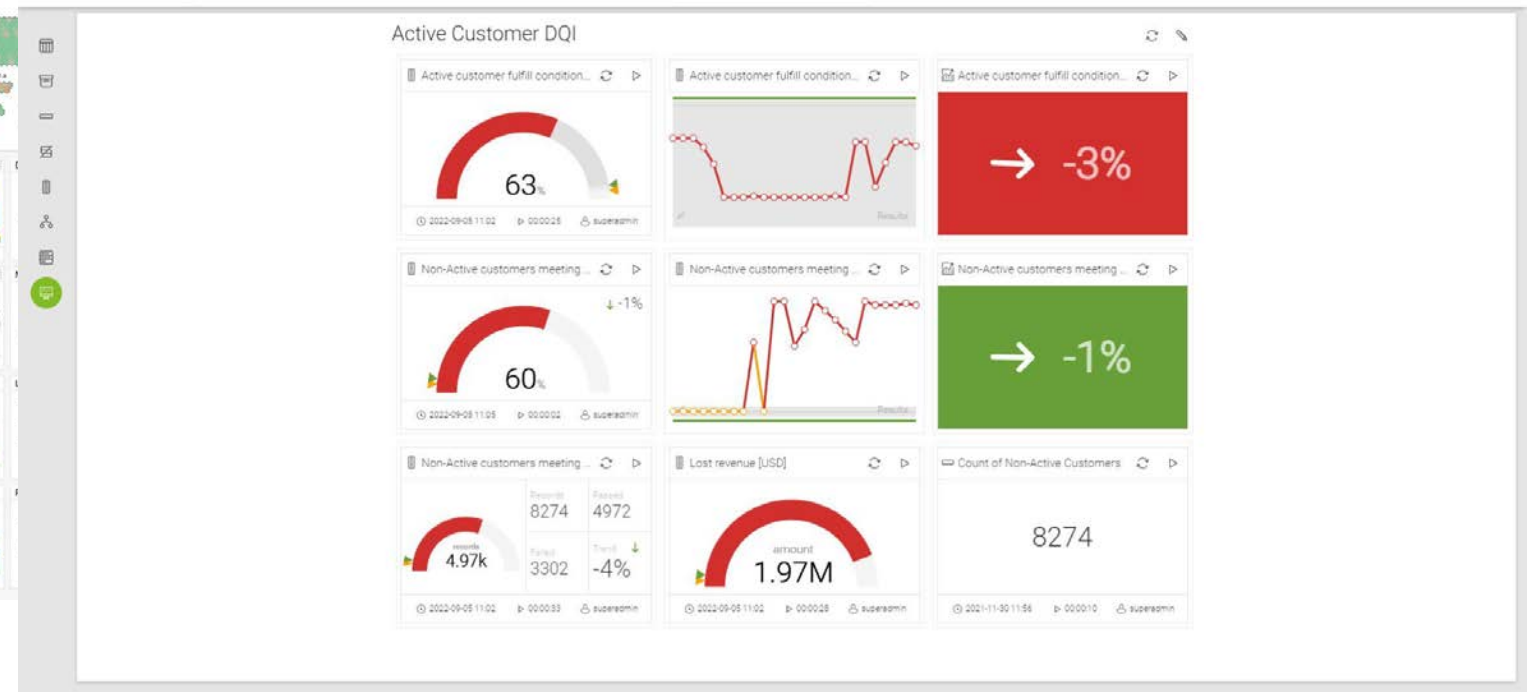
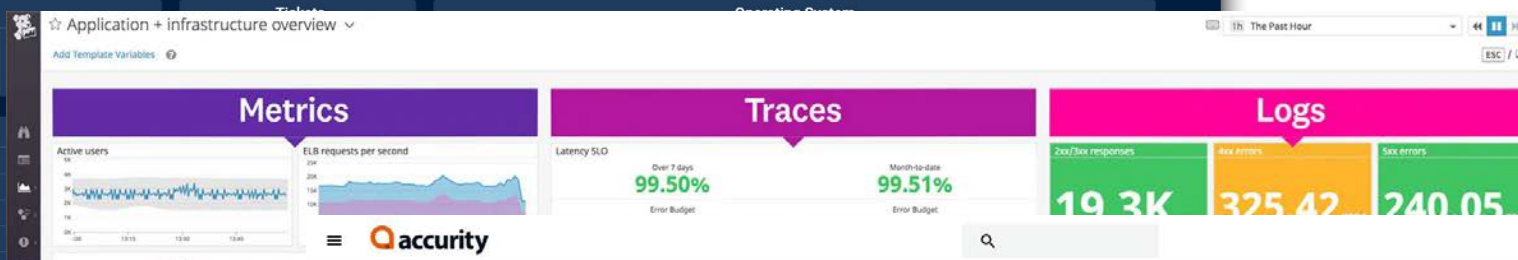
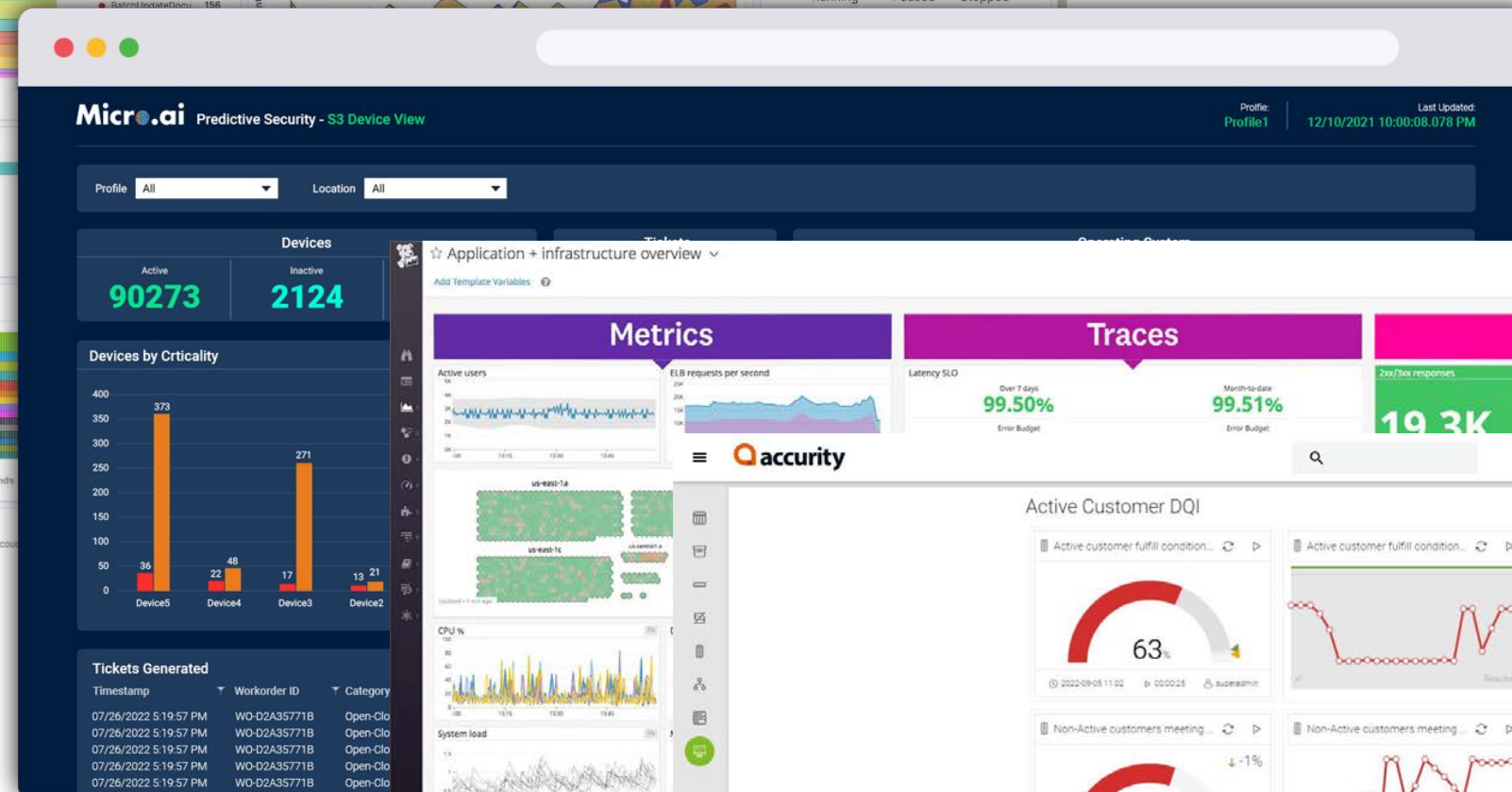
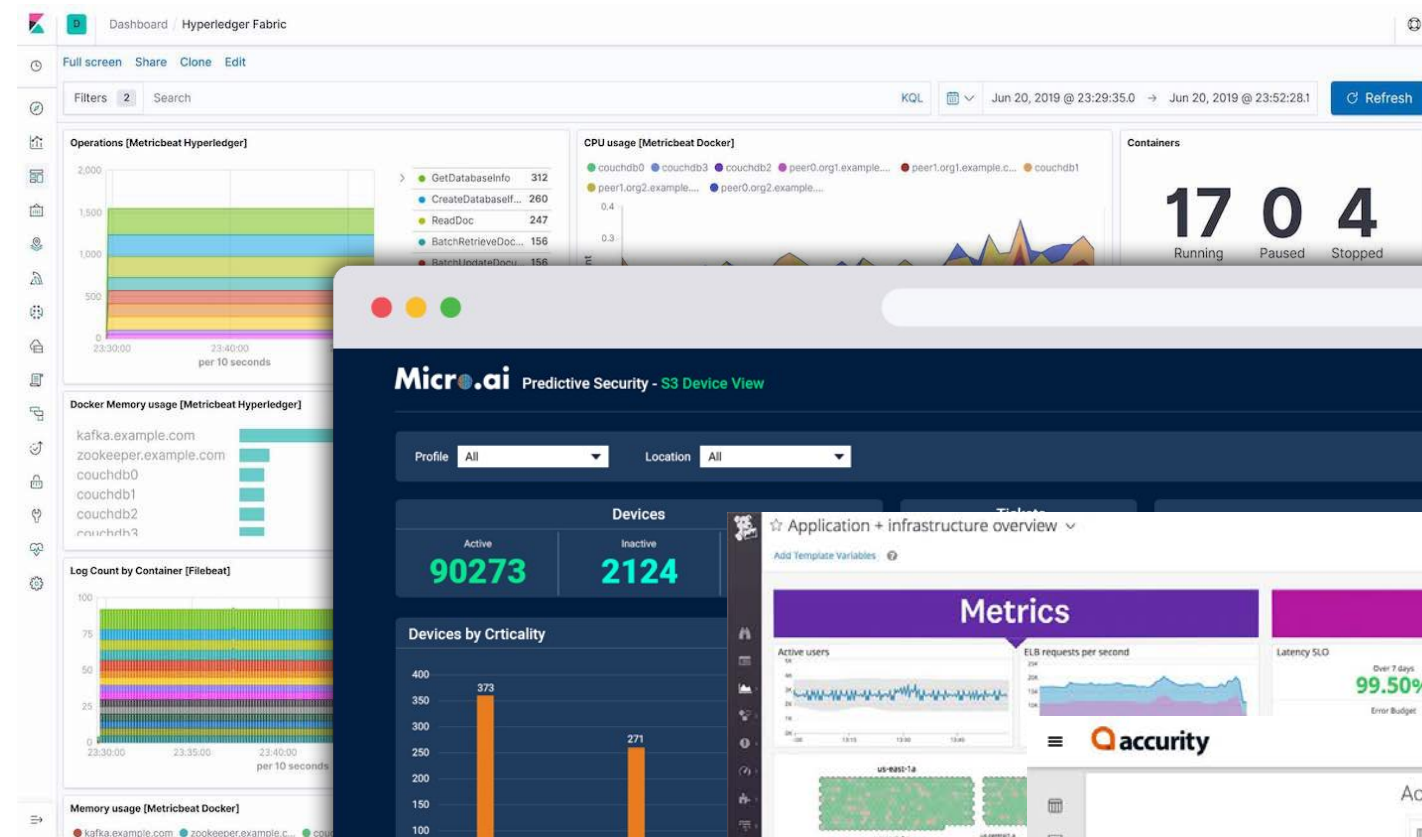
by: Siarhei Misko

Logging

by: Siarhei Misko

Optimize!

by: Siarhei Misko





Aa  



Intro

by: Siarhei Misko

Observability

by: Siarhei Misko

Observability in iOS

by: Siarhei Misko

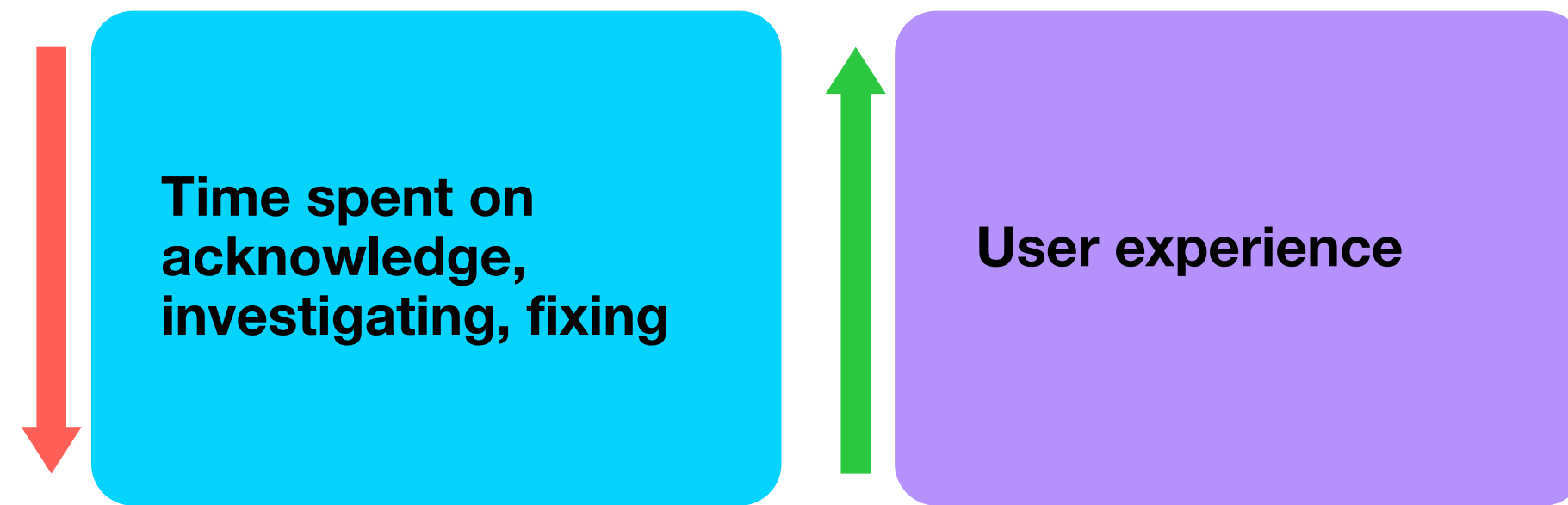
Logging

by: Siarhei Misko

Optimize!

by: Siarhei Misko

Observability value





Aa



Intro

by: Siarhei Misko

Observability

by: Siarhei Misko

Observability in iOS

by: Siarhei Misko

Logging

by: Siarhei Misko

Optimize!

by: Siarhei Misko

Observability value

**Learn about problem
before it becomes
critical**

**Get complete and
detailed information**



Aa  



Intro

by: Siarhei Misko

Observability

by: Siarhei Misko

Observability in iOS

by: Siarhei Misko

Logging

by: Siarhei Misko

Optimize!

by: Siarhei Misko



Aa   



Intro

by: Siarhei Misko

Observability

by: Siarhei Misko

Observability in iOS

by: Siarhei Misko

Logging

by: Siarhei Misko

Optimize!

by: Siarhei Misko

iOS project

Case

- iOS app without any monitoring, crashlytics, etc.
- New version tested and released
- Team received large number of complaints about possible crashes
- QA can't reproduce issue in their environment



Intro

by: Siarhei Misko

Observability

by: Siarhei Misko

Observability in iOS

by: Siarhei Misko

Logging

by: Siarhei Misko

Optimize!

by: Siarhei Misko

iOS project

Case

- iOS app without any monitoring, crashlytics, etc.
- New version tested and released
- Team received large number of complaints about possible crashes
- QA can't reproduce issue in their environment

Solution

- Blindly fix issue
- Release new version
- *Hope it works* 🙄



Aa [font settings icon] [globe icon]



Intro

by: Siarhei Misko

Observability

by: Siarhei Misko

Observability in iOS

by: Siarhei Misko

Logging

by: Siarhei Misko

Optimize!

by: Siarhei Misko

Solution:

connect crash analytics service





Aa [font settings icon] [globe icon]



Intro

by: Siarhei Misko

Observability

by: Siarhei Misko

Observability in iOS

by: Siarhei Misko

Logging

by: Siarhei Misko

Optimize!

by: Siarhei Misko

The screenshot shows the Xcode interface with the 'Crashes' tab selected. The main area displays a crash report for 'Forest Explorer: ActivityDetailViewController.tableView(_:didSele...' on 'Thread 0'. The stack trace includes:

- 0 Forest Explorer ActivityDetailViewController.table...
- 1 Forest Explorer @objc ActivityDetailViewController.ta...
- 2 Forest Explorer -[UITableView_selectRowAtIndexPat...
- 12 GraphicsServices GSEventRunModal
- 13 Forest Explorer UIApplicationMain
- 14 Forest Explorer main
- 15 libdyld.dylib start

On the right, the 'Crash Information' panel shows:

- Crash Information: Forest Explorer: ActivityD... (61 devices)
- Crash Log Details: Binary Forest Explorer 4.5.4 (454...), Thread 0, iOS 13.3 (17C54), Device iPhone 8 (Model A1863, A1...)
- Last 2 Weeks: 61 TOTAL crashes (3% iPad, 97% iPhone)



Intro

by: Siarhei Misko

Observability

by: Siarhei Misko

Observability in iOS

by: Siarhei Misko

Logging

by: Siarhei Misko

Optimize!

by: Siarhei Misko

Crashlytics iOS Friendly Eats

Filter Versions + All events Last 7 days Oct 25 - Oct 31, 2017 (PST)

Crash-free statistics
Crash-free users: **98.61%** (+0.5%)

Event trends

Crashes	Users affected	Non-fatals	Users affected
203 (-17.7%)	232 (+6.9%)	1402	403

Issues (10)

Issues	Details	Versions	Count	Users
FiltersViewController.swift - Line 75 FiltersViewController.price(from : String) -> Int? iOS 11	Crash	1.0 - 1.5	89	52
FiltersViewController.swift - Line 96 FiltersViewController.clearFilters() -> ().closure #1 Insight 6	Crash	1.0 - 1.5	54	38
Restaurant.swift - Line 65 RestaurantViewController.doneButtonPressed(Any) -> ()	Crash	1.0 - 1.1	32	25
StarsView.swift - Line 39 StarsViewController.doneButtonPressed(Any) -> () 1.8%	Crash	1.4	28	19



Aa  



Intro

by: Siarhei Misko

Observability

by: Siarhei Misko

Observability in iOS

by: Siarhei Misko

Logging

by: Siarhei Misko

Optimize!

by: Siarhei Misko

$O(n!)$





Intro

by: Siarhei Misko

Observability

by: Siarhei Misko

Observability in iOS

by: Siarhei Misko

Logging

by: Siarhei Misko

Optimize!

by: Siarhei Misko

```
[
  {
    "feature_name": "quick_payments",
    "enabled": true,
    "payload": null
  },
  {
    "feature_name": "contact_list_payment",
    "enabled": true,
    "payload": {
      "mandatory_string_field_1": "string_value",
      // missing mandatory field 2 here
      "mandatory_string_field_3": "string_value",
      "mandatory_int_field_4": "345" // miscast here, should be a number
    }
  }
]
```



Aa   



Intro

by: Siarhei Misko

Observability

by: Siarhei Misko

Observability in iOS

by: Siarhei Misko

Logging

by: Siarhei Misko

Optimize!

by: Siarhei Misko

Conditions to be taken into account

Different scenarios

App settings

Different localizations

All network conditions

All supported iOS devices
All supported iPads
All supported macs with Apple Silicon

Feature flags



Aa



Intro

by: Siarhei Misko

Observability

by: Siarhei Misko

Observability in iOS

by: Siarhei Misko

Logging

by: Siarhei Misko

Optimize!

by: Siarhei Misko

1

Important user stories



Aa   



Intro

by: Siarhei Misko

Observability

by: Siarhei Misko

Observability in iOS

by: Siarhei Misko

Logging

by: Siarhei Misko

Optimize!

by: Siarhei Misko

2

Flows, features, stories dependent on dynamic changing arguments



Aa   



Intro

by: Siarhei Misko

Observability

by: Siarhei Misko

Observability in iOS

by: Siarhei Misko

Logging

by: Siarhei Misko

Optimize!

by: Siarhei Misko

3

Flows, features, stories dependent on external systems



Intro

by: Siarhei Misko

Observability

by: Siarhei Misko

Observability in iOS

by: Siarhei Misko

Logging

by: Siarhei Misko

Optimize!

by: Siarhei Misko

TIME TO STUDY





Aa   



Intro

by: Siarhei Misko

Observability

by: Siarhei Misko

Observability in iOS

by: Siarhei Misko

Logging

by: Siarhei Misko

Optimize!

by: Siarhei Misko

LOGGING IN IOS APPLICATIONS

Intro

by: Siarhei Misko

Observability

by: Siarhei Misko

Observability in iOS

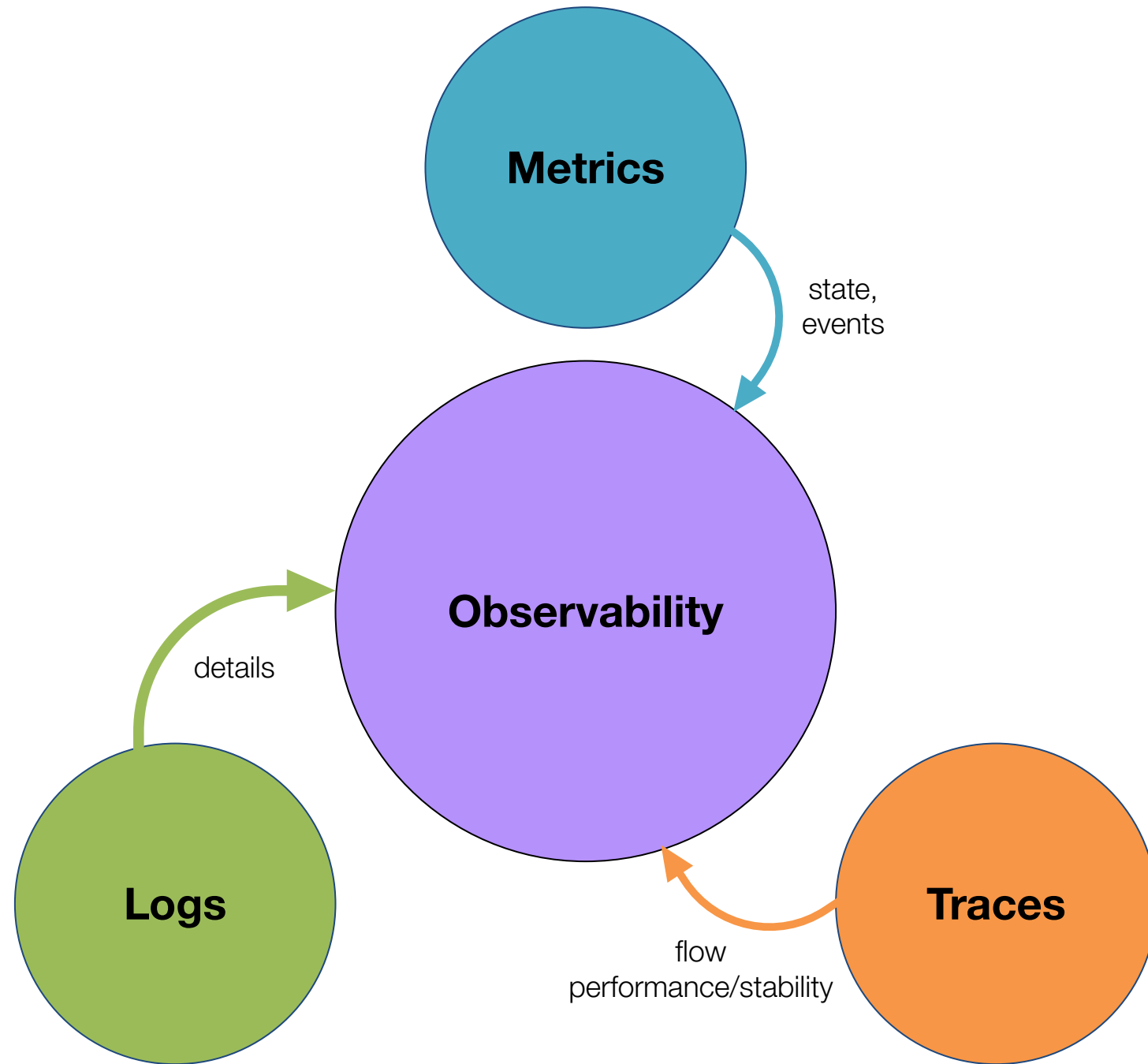
by: Siarhei Misko

Logging

by: Siarhei Misko

Optimize!

by: Siarhei Misko





Aa   



Intro

by: Siarhei Misko

Observability

by: Siarhei Misko

Observability in iOS

by: Siarhei Misko

Logging

by: Siarhei Misko

Optimize!

by: Siarhei Misko

Logs delivery

Live (online)

On demand

- Backend controlled
- Manual

By trigger



Aa



Intro

by: Siarhei Misko

Observability

by: Siarhei Misko

Observability in iOS

by: Siarhei Misko

Logging

by: Siarhei Misko

Optimize!

by: Siarhei Misko

Logs collecting

Continious

On demand

By trigger



Aa   



Intro

by: Siarhei Misko

Observability

by: Siarhei Misko

Observability in iOS

by: Siarhei Misko

Logging

by: Siarhei Misko

Optimize!

by: Siarhei Misko

Continuous logs collecting

- Avoid recording too many logs at once to prevent overloading the disk with read/write operations.
- Implement log file rotation to prevent them from becoming too large.
- Carefully manage logging levels



Aa  



Intro

by: Siarhei Misko

Observability

by: Siarhei Misko

Observability in iOS

by: Siarhei Misko

Logging

by: Siarhei Misko

Optimize!

by: Siarhei Misko

Logs collecting

Continious

On demand

By trigger



Aa



Intro

by: Siarhei Misko

Observability

by: Siarhei Misko

Observability in iOS

by: Siarhei Misko

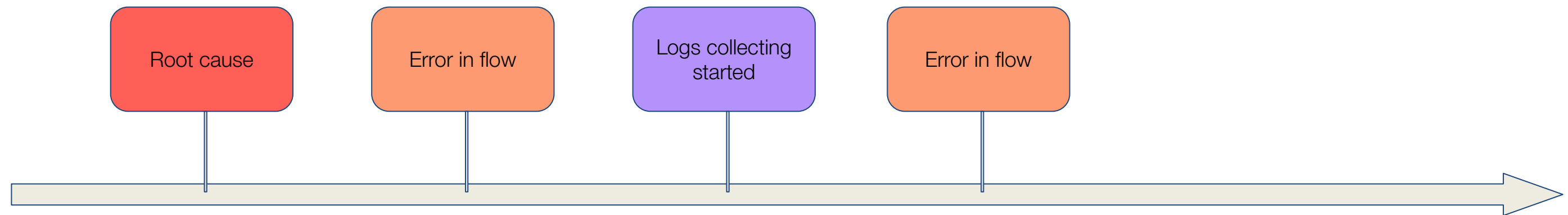
Logging

by: Siarhei Misko

Optimize!

by: Siarhei Misko

Logs collecting





Intro

by: Siarhei Misko

Observability

by: Siarhei Misko

Observability in iOS

by: Siarhei Misko

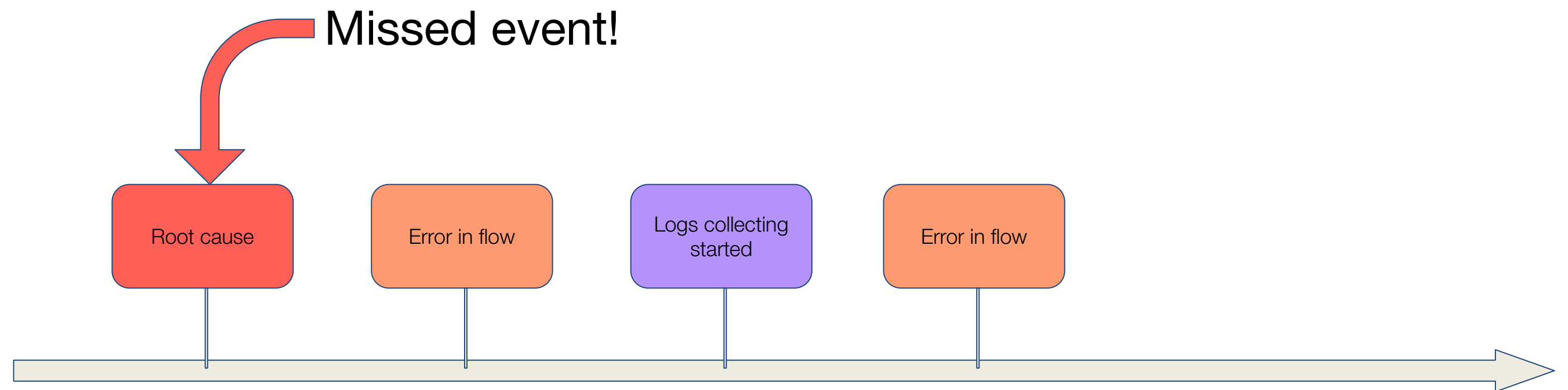
Logging

by: Siarhei Misko

Optimize!

by: Siarhei Misko

Logs collecting





Aa



Intro

by: Siarhei Misko

Observability

by: Siarhei Misko

Observability in iOS

by: Siarhei Misko

Logging

by: Siarhei Misko

Optimize!

by: Siarhei Misko

Combined approach

- Debug
- Info
- Notice
- Warning
- Error
- Critical



Aa



Intro

by: Siarhei Misko

Observability

by: Siarhei Misko

Observability in iOS

by: Siarhei Misko

Logging

by: Siarhei Misko

Optimize!

by: Siarhei Misko

Combined approach

- Debug
 - Info
 - Notice
 - Warning
 - Error
 - Critical
- Continuous**



Aa



Intro

by: Siarhei Misko

Observability

by: Siarhei Misko

Observability in iOS

by: Siarhei Misko

Logging

by: Siarhei Misko

Optimize!

by: Siarhei Misko

Combined approach

- Debug
 - Info
 - Notice
 - Warning
 - Error
 - Critical
- Configurable**
- Continuous**



Aa



Intro

by: Siarhei Misko

Observability

by: Siarhei Misko

Observability in iOS

by: Siarhei Misko

Logging

by: Siarhei Misko

Optimize!

by: Siarhei Misko

Collect logs in iOS

- OSLog
- Textfile
- Firebase *(for starters)*



Aa   



Intro

by: Siarhei Misko

Observability

by: Siarhei Misko

Observability in iOS

by: Siarhei Misko

Logging

by: Siarhei Misko

Optimize!

by: Siarhei Misko

Collect logs in iOS with OSLog

- Connect the phone to a computer and view the console.
- Collect a sysdiagnose of the entire device, which takes up to 15 minutes and results in a 200+ MB archive.
- Programmatically from the app, but only for current pld

Intro

by: Siarhei Misko

Observability

by: Siarhei Misko

Observability in iOS

by: Siarhei Misko

Logging

by: Siarhei Misko

Optimize!

by: Siarhei Misko

Collect logs in iOS to text file

- Start with open-source frameworks

Logging libraries

Showing projects tagged as Logging

SwiftyBeaver

★▶ 9.5 ❤▶ 6.9 L4 </> Swift

Convenient & secure logging during development & release in Swift 4 & 5



XCGLogger

★▶ 9.3 ❤▶ 7.6 L3 </> Swift

A debug log framework for use in Swift projects. Allows you to log details to the console (and optionally a file), just like you would have with NSLog() or print(), but with additional information, such as the date, function name, filename and line number.

Dotzu

★▶ 8.3 ❤▶ 0.0 L5 </> Swift

DISCONTINUED. Debugger tool for iOS developer. Display logs, network request, device informations, crash logs while using the app.

TinyConsole

★▶ 8.2 ❤▶ 0.0 L5 </> Swift

TinyConsole is a micro-console that can help you log and display information inside an iOS

```
// add log destinations. at least one is needed!
let console = ConsoleDestination() // log to Xcode Console
let file = FileDestination() // log to default swiftybeaver.log file

// use custom format and set console output to short time, log level & message
console.format = "$DHH:mm:ss$ $L $M"
// or use this for JSON output: console.format = "$J"

// In Xcode 15, specifying the logging method as .logger to display color, subsystem, and ca
console.logPrintWay = .logger(subsystem: "Main", category: "UI")
// If you prefer not to use the OSLog API, you can use print instead.
// console.logPrintWay = .print

// add the destinations to SwiftyBeaver
log.addDestination(console)
log.addDestination(file)

// Now let's log!
log.verbose("not so important") // prio 1, VERBOSE in silver
log.debug("something to debug") // prio 2, DEBUG in green
log.info("a nice information") // prio 3, INFO in blue
log.warning("oh no, that won't be good") // prio 4, WARNING in yellow
log.error("ouch, an error did occur!") // prio 5, ERROR in red

// log anything!
log.verbose(123)
log.info(-123.45678)
log.warning(Date())
log.error(["I", "like", "logs!"])
log.error(["name": "Mr Beaver", "address": "7 Beaver Lodge"])

// optionally add context to a log message
console.format = "$L: $M $X"
log.debug("age", context: 123) // "DEBUG: age 123"
log.info("my data", context: [1, "a", 2]) // "INFO: my data [1, \"a\", 2]"
```

Intro

by: Siarhei Misko

Observability

by: Siarhei Misko

Observability in iOS

by: Siarhei Misko

Logging

by: Siarhei Misko

Optimize!

by: Siarhei Misko

Collect logs in iOS with Firebase Crashlytics

Event summary 1.5.3 (2500) iOS+ iOS 17.5.1 iPhone 14 Aug 5, 2024, 7:20:00 PM

Stack trace Keys Logs & Breadcrumbs Data Rollouts

TXT

Crashed: com.apple.main-thread
EXC_BREAKPOINT 0x0000000195fe8924

0	libdispatch.dylib	_dispatch_queue_xref_dispose.cold.2 + 36
1	libdispatch.dylib	_dispatch_lane_suspend + 62

Event summary 1.5.3 (2500) iOS+ iOS 17.5.1 iPhone 14 Aug 5, 2024, 7:20:00 PM

Stack trace Keys Logs & Breadcrumbs Data Rollouts

Device	Operating System	Crash
Model: iPhone 14 Orientation: Portrait RAM free: 307.19 MB Disk free: 80.95 GB	Version: 17.5.1 Orientation: Portrait Jailbroken: No	Date: Aug 5, 2024, 7:20:00 PM App version: 1.5.3 (2500)



Intro

by: Siarhei Misko

Observability

by: Siarhei Misko

Observability in iOS

by: Siarhei Misko

Logging

by: Siarhei Misko

Optimize!

by: Siarhei Misko

Collect logs in iOS with Firebase Crashlytics

Stack trace

Keys

Logs & Breadcrumbs

Data

Rollouts

Filter keys

Key

Value for this event

crash_info_entry_0

BUG IN CLIENT OF LIBDISPATCH: Release of an inactive object

isPremium

0



Intro

by: Siarhei Misko

Observability

by: Siarhei Misko

Observability in iOS

by: Siarhei Misko

Logging

by: Siarhei Misko

Optimize!

by: Siarhei Misko

Collect logs in iOS with Firebase Crashlytics

Event summary 1.5.3 (2500) iOS+ iOS 17.5.1 iPhone 14 Aug 5, 2024, 7:20:00 PM

Stack trace Keys Logs & Breadcrumbs Data Rollouts

Filter logs Download .log

#	Time (newest first)	Source	Log
1	07:15:09.423 PM		Show launch animation dispatch leave called



Intro

by: Siarhei Misko

Observability

by: Siarhei Misko

Observability in iOS

by: Siarhei Misko

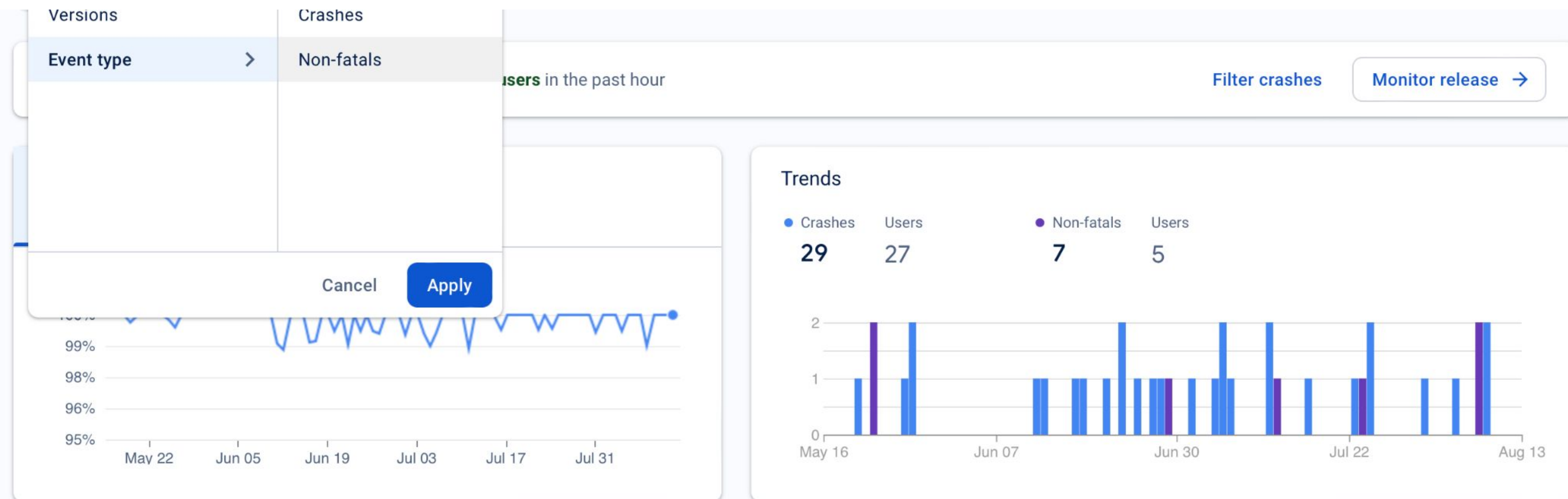
Logging

by: Siarhei Misko

Optimize!

by: Siarhei Misko

Collect logs in iOS with Firebase Crashlytics





Aa   



Intro

by: Siarhei Misko

Observability

by: Siarhei Misko

Observability in iOS

by: Siarhei Misko

Logging

by: Siarhei Misko

Optimize!

by: Siarhei Misko

Collect logs in iOS with Firebase Crashlytics

- Data limit
- No custom queries



Aa



Intro

by: Siarhei Misko

Observability

by: Siarhei Misko

Observability in iOS

by: Siarhei Misko

Logging

by: Siarhei Misko

Optimize!

by: Siarhei Misko

General Advices for Writing Logs to a Text File



Intro

by: Siarhei Misko

Observability

by: Siarhei Misko

Observability in iOS

by: Siarhei Misko

Logging

by: Siarhei Misko

Optimize!

by: Siarhei Misko

General Advices for Writing Logs to a Text File

- **Debug** - very detailed information, mostly for developers during troubleshooting
- **Info** - general operational messages about the application's progress
- **Notice** - normal but significant events, slightly above Info in importance
- **Warning** - indications of potential issues or risks that aren't yet errors
- **Error** - serious issues that affect functionality but allow the application to continue running
- **Critical** - severe errors causing partial or complete application failure



Intro

by: Siarhei Misko

Observability

by: Siarhei Misko

Observability in iOS

by: Siarhei Misko

Logging

by: Siarhei Misko

Optimize!

by: Siarhei Misko

General Advices for Writing Logs to a Text File

```
10 final class Logger {
11
12     static func debug(subdomain: CustomStringConvertible = "",
13                       _ msg: String,
14                       file: String = #file,
15                       function: String = #function,
16                       line: Int = #line) {
17         // some implementation of logger here
18     }
19 }
20
21 final class ViewController: UIViewController {
22
23     override func viewDidLoad() {
24         super.viewDidLoad()
25
26         #if LOGGING_ENABLED
27         Logger.debug("ViewController viewDidLoad")
28         #endif
29     }
30
31     override func viewWillAppear(_ animated: Bool) {
32         super.viewWillAppear(animated)
33
34         #if LOGGING_ENABLED
35         Logger.debug("ViewController viewWillAppear")
36         #endif
37     }
38 }
```



Intro

by: Siarhei Misko

Observability

by: Siarhei Misko

Observability in iOS

by: Siarhei Misko

Logging

by: Siarhei Misko

Optimize!

by: Siarhei Misko

General Advices for Writing Logs to a Text File

```
10 final class Logger {
11
12     #if LOGGING_ENABLED
13     static func debug(subdomain: CustomStringConvertible = "",
14                       _ msg: String,
15                       file: String = #file,
16                       function: String = #function,
17                       line: Int = #line) {
18         // some implementation of logger here
19     }
20     #else
21     static func debug(subdomain: CustomStringConvertible = "",
22                       _ msg: String,
23                       file: String = #file,
24                       function: String = #function,
25                       line: Int = #line) {} // EMPTY HERE
26     #endif
27 }
28
29 final class ViewController: UIViewController {
30
31     override func viewDidLoad() {
32         super.viewDidLoad()
33
34         Logger.debug("ViewController viewDidLoad")
35     }
36
37     override func viewWillAppear(_ animated: Bool) {
38         super.viewWillAppear(animated)
39
40         Logger.debug("ViewController viewWillAppear")
41     }
42 }
```



Intro

by: Siarhei Misko

Observability

by: Siarhei Misko

Observability in iOS

by: Siarhei Misko

Logging

by: Siarhei Misko

Optimize!

by: Siarhei Misko

General Advices for Writing Logs to a Text File

```
10 final class Logger {
11
12     #if LOGGING_ENABLED
13     static func debug(subdomain: CustomStringConvertible = "",
14                       msg: @autoclosure () -> CustomStringConvertible,
15                       file: String = #file,
16                       function: String = #function,
17                       line: Int = #line) {
18         // some implementation of logger here
19     }
20     #else
21     static func debug(subdomain: CustomStringConvertible = "",
22                       _ msg: @autoclosure () -> CustomStringConvertible,
23                       file: String = #file,
24                       function: String = #function,
25                       line: Int = #line) {} // EMPTY HERE
26     #endif
27 }
28
29 final class ViewController: UIViewController {
30
31     override func viewDidLoad() {
32         super.viewDidLoad()
33
34         Logger.debug("ViewController viewDidLoad")
35     }
36
37     override func viewWillAppear(_ animated: Bool) {
38         super.viewWillAppear(animated)
39
40         Logger.debug("ViewController viewWillAppear")
41     }
42 }
```



Aa



Intro

by: Siarhei Misko

Observability

by: Siarhei Misko

Observability in iOS

by: Siarhei Misko

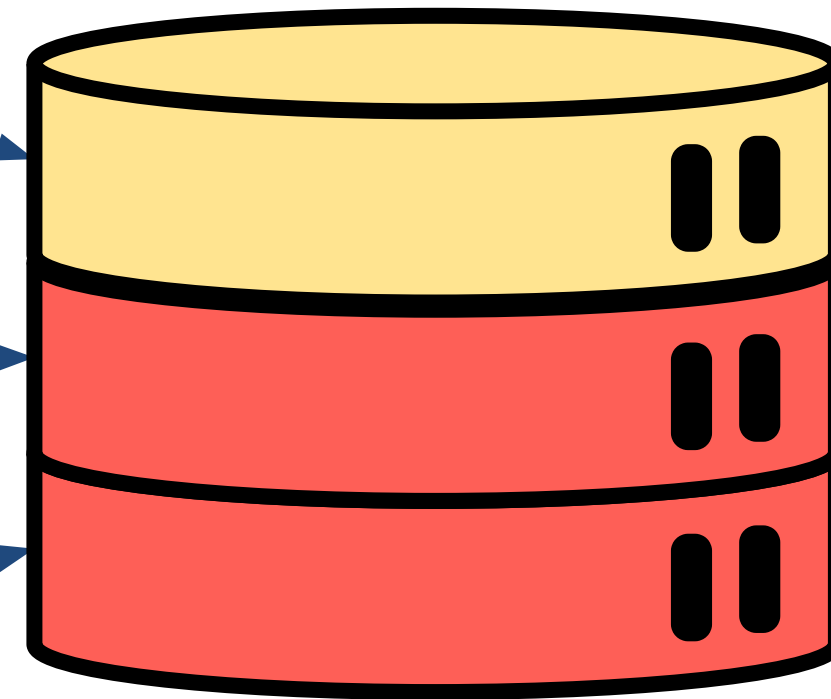
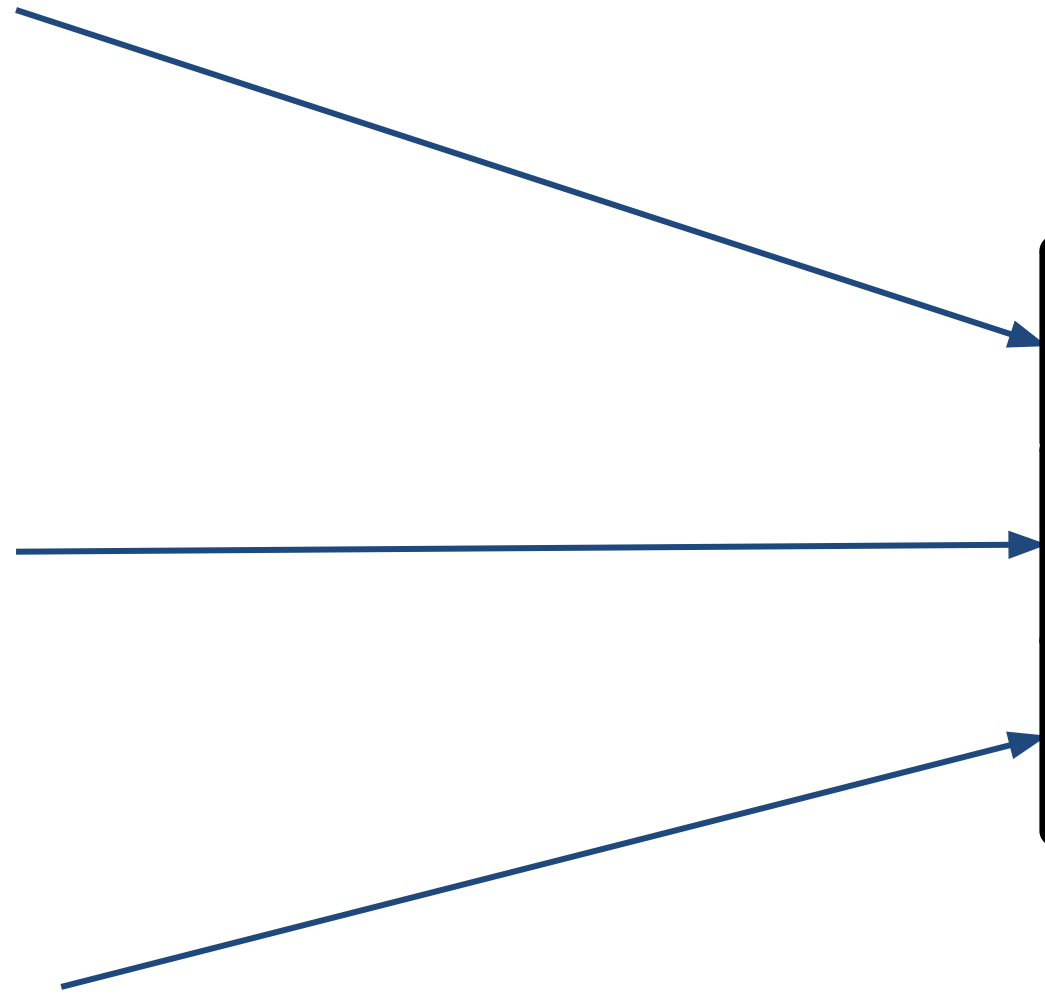
Logging

by: Siarhei Misko

Optimize!

by: Siarhei Misko

Logs size





Aa   



Intro

by: Siarhei Misko

Observability

by: Siarhei Misko

Observability in iOS

by: Siarhei Misko

Logging

by: Siarhei Misko

Optimize!

by: Siarhei Misko

Send state instead of logs

- What screens viewed
- Which API requests made
- Settings values
- Business logic flags
- Feature flags



Intro

by: Siarhei Misko

Observability

by: Siarhei Misko

Observability in iOS

by: Siarhei Misko

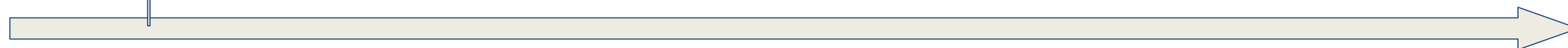
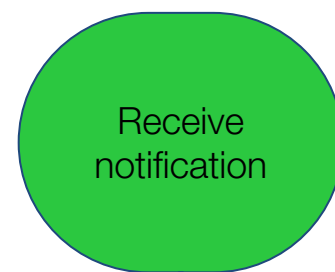
Logging

by: Siarhei Misko

Optimize!

by: Siarhei Misko

Notification Service Extension





Intro

by: Siarhei Misko

Observability

by: Siarhei Misko

Observability in iOS

by: Siarhei Misko

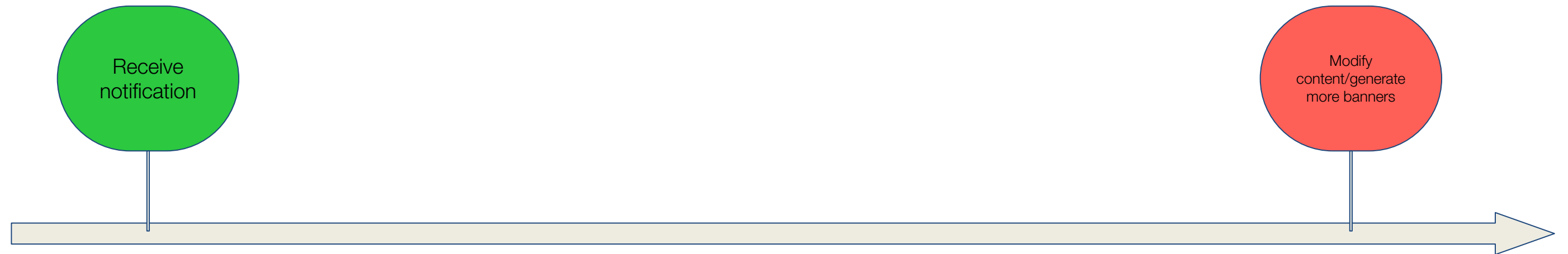
Logging

by: Siarhei Misko

Optimize!

by: Siarhei Misko

Notification Service Extension





Intro

by: Siarhei Misko

Observability

by: Siarhei Misko

Observability in iOS

by: Siarhei Misko

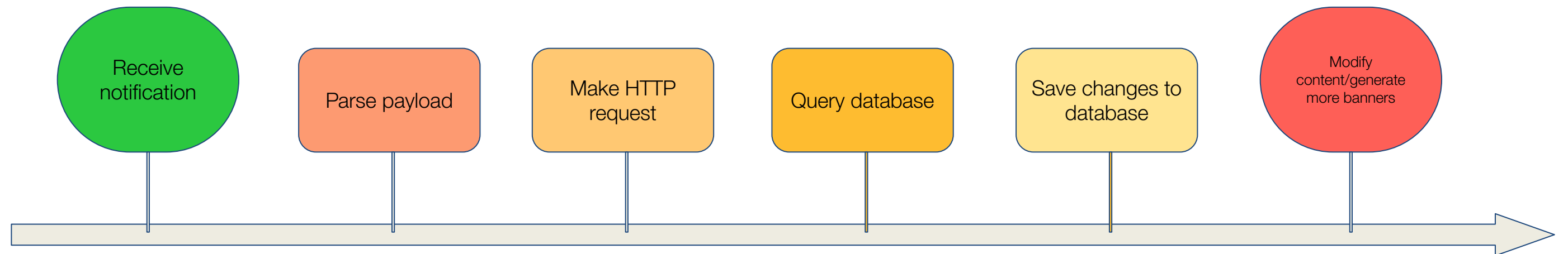
Logging

by: Siarhei Misko

Optimize!

by: Siarhei Misko

Notification Service Extension



Intro

by: Siarhei Misko

Observability

by: Siarhei Misko

Observability in iOS

by: Siarhei Misko

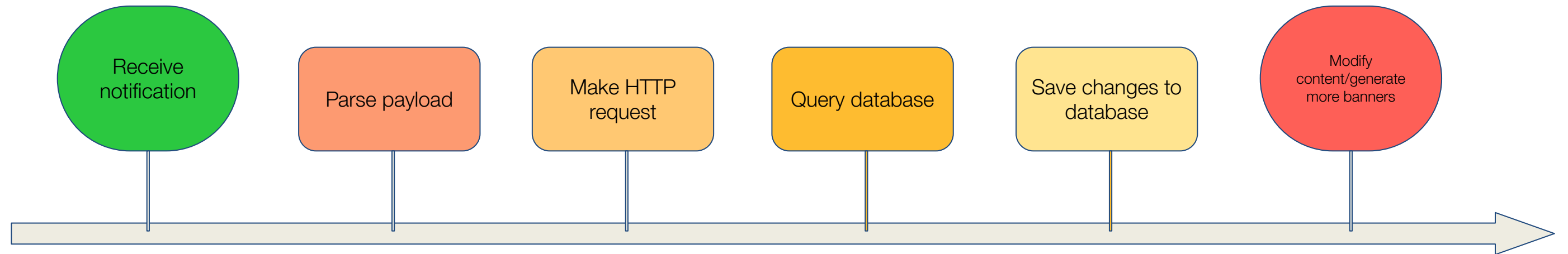
Logging

by: Siarhei Misko

Optimize!

by: Siarhei Misko

Notification Service Extension





Aa



Intro

by: Siarhei Misko

Observability

by: Siarhei Misko

Observability in iOS

by: Siarhei Misko

Logging

by: Siarhei Misko

Optimize!

by: Siarhei Misko

Notification Service Extension steps

```
44 struct Tasks: OptionSet {
45     let rawValue: Int
46
47     static let receiveNotification = Tasks(rawValue: 1 << 0)
48     static let parsePayload       = Tasks(rawValue: 1 << 1)
49     static let makeHTTPRequest    = Tasks(rawValue: 1 << 2)
50     static let queryDatabase      = Tasks(rawValue: 1 << 3)
51     static let saveChangesToDB    = Tasks(rawValue: 1 << 4)
52     static let generateBanners    = Tasks(rawValue: 1 << 5)
53     static let finish             = Tasks(rawValue: 1 << 6)
54 }
```



Intro

by: Siarhei Misko

Observability

by: Siarhei Misko

Observability in iOS

by: Siarhei Misko

Logging

by: Siarhei Misko

Optimize!

by: Siarhei Misko

Notification Service Extension steps

```
57 func checkPreviousRun(tasks: Tasks) {  
58     guard tasks.contains(.finish) else {  
59         report(msg: "previous run not finished, \(tasks)")  
60         return  
61     }  
62  
63     // other checks  
64 }
```



Intro

by: Siarhei Misko

Observability

by: Siarhei Misko

Observability in iOS

by: Siarhei Misko

Logging

by: Siarhei Misko

Optimize!

by: Siarhei Misko

“
**It's important to
follow established
patterns**

with an emphasis on adhering to SOLID principles



Aa



Intro

by: Siarhei Misko

Observability

by: Siarhei Misko

Observability in iOS

by: Siarhei Misko

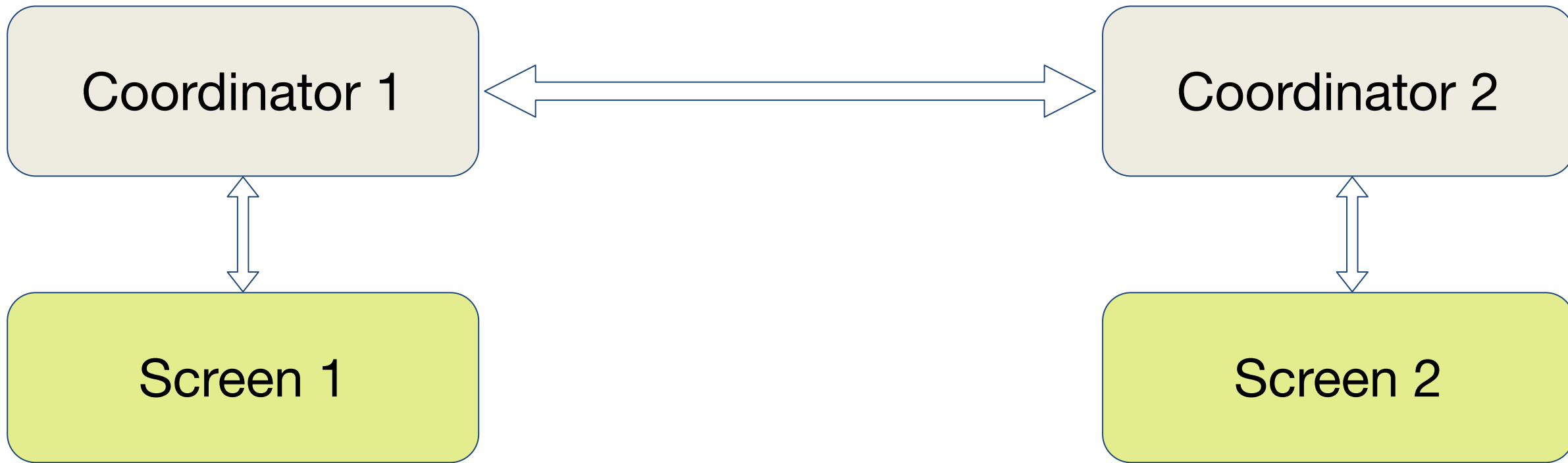
Logging

by: Siarhei Misko

Optimize!

by: Siarhei Misko

Coordinator pattern





Intro

by: Siarhei Misko

Observability

by: Siarhei Misko

Observability in iOS

by: Siarhei Misko

Logging

by: Siarhei Misko

Optimize!

by: Siarhei Misko

Coordinator pattern

Registration -> Enter Code -> Main Screen -> More Screen -> Settings



Intro

by: Siarhei Misko

Observability

by: Siarhei Misko

Observability in iOS

by: Siarhei Misko

Logging

by: Siarhei Misko

Optimize!

by: Siarhei Misko

Coordinator pattern

Registration -> Enter Code -> Main Screen -> More Screen -> Settings



R1 -> E1 -> M1 -> M2 -> M3



Aa  



Intro

by: Siarhei Misko

Observability

by: Siarhei Misko

Observability in iOS

by: Siarhei Misko

Logging

by: Siarhei Misko

Optimize!

by: Siarhei Misko

Coordinator pattern

```
45 func applicationWillTerminate(_ application: UIApplication) {  
46     settingsStorage.lastScreenBeforeTerminate = rootCoordinator.presentedScreenName  
47 }  
48
```



Aa



Intro

by: Siarhei Misko

Observability

by: Siarhei Misko

Observability in iOS

by: Siarhei Misko

Logging

by: Siarhei Misko

Optimize!

by: Siarhei Misko

Coordinator pattern

```
35 func application(_ application: UIApplication,
36                 didFinishLaunchingWithOptions launchOptions: [UIApplication.LaunchOptionsKey: Any]?) -> Bool {
37
38     if let lastScreenBeforeTerminate = settingsStorage.lastScreenBeforeTerminate {
39         metricsService.track(.lastScreenBeforeTerminate, value: lastScreenBeforeTerminate)
40     }
41
42     return true
43 }
44
45 func applicationWillTerminate(_ application: UIApplication) {
46     settingsStorage.lastScreenBeforeTerminate = rootCoordinator.presentedScreenName
47 }
48
```



Aa  



Intro

by: Siarhei Misko

Observability

by: Siarhei Misko

Observability in iOS

by: Siarhei Misko

Logging

by: Siarhei Misko

Optimize!

by: Siarhei Misko

Coordinator pattern

- Track App Start time
- Track User Story time



Aa



Intro

by: Siarhei Misko

Observability

by: Siarhei Misko

Observability in iOS

by: Siarhei Misko

Logging

by: Siarhei Misko

Optimize!

by: Siarhei Misko

Network/API service

```
57 func executeHttpRequest(method: HTTPMethod,  
58     url: URL,  
59     headers: [String: String]?,  
60     body: Data?,  
61     completion: ((Result<Data, Error>) -> Void)?) {  
62     // 1. implementation of making HTTP Request  
63  
64     // 2. handling response or error  
65  
66     metricsService.trackHttpRequest(url: url, elapsedTime: currentTimeSinceStart)  
67 }
```



Aa



Intro

by: Siarhei Misko

Observability

by: Siarhei Misko

Observability in iOS

by: Siarhei Misko

Logging

by: Siarhei Misko

Optimize!

by: Siarhei Misko

Annoyed tap





Intro

by: Siarhei Misko

Observability

by: Siarhei Misko

Observability in iOS

by: Siarhei Misko

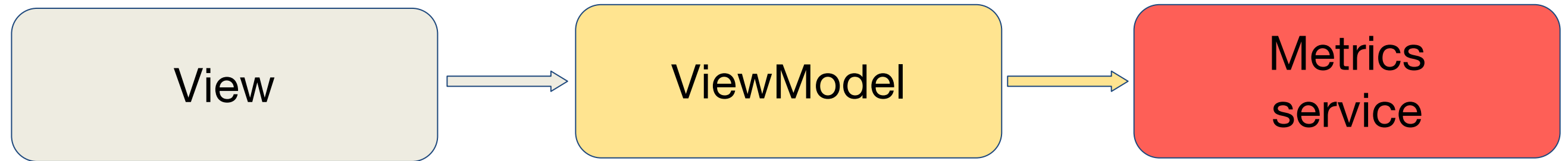
Logging

by: Siarhei Misko

Optimize!

by: Siarhei Misko

Annoyed tap





Aa



Intro

by: Siarhei Misko

Observability

by: Siarhei Misko

Observability in iOS

by: Siarhei Misko

Logging

by: Siarhei Misko

Optimize!

by: Siarhei Misko

Thank you!

iOS Team Lead

siarheimisko@gmail.com

