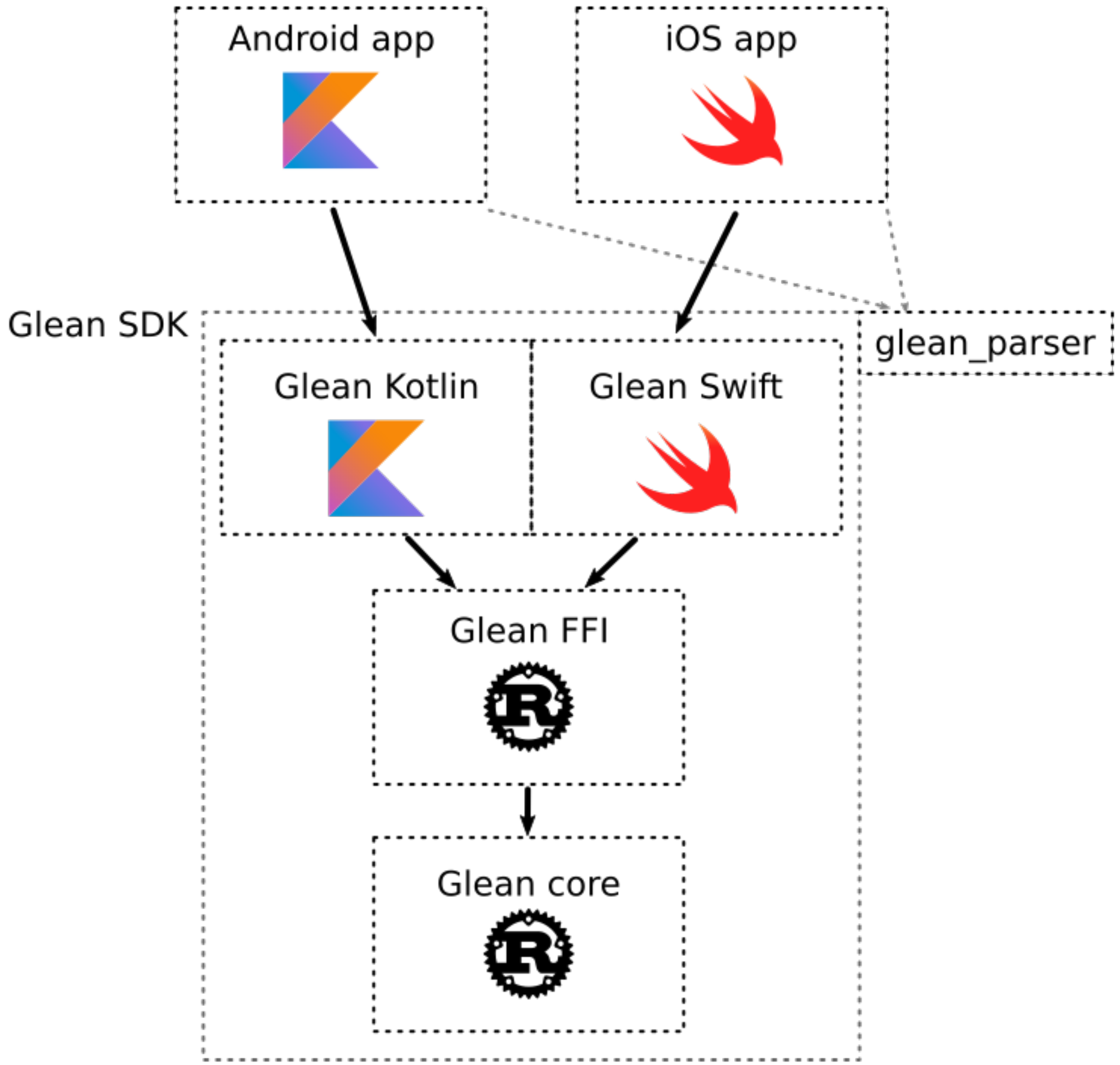
GLEAN

telemetry for humans

The Glean UniFFI migration

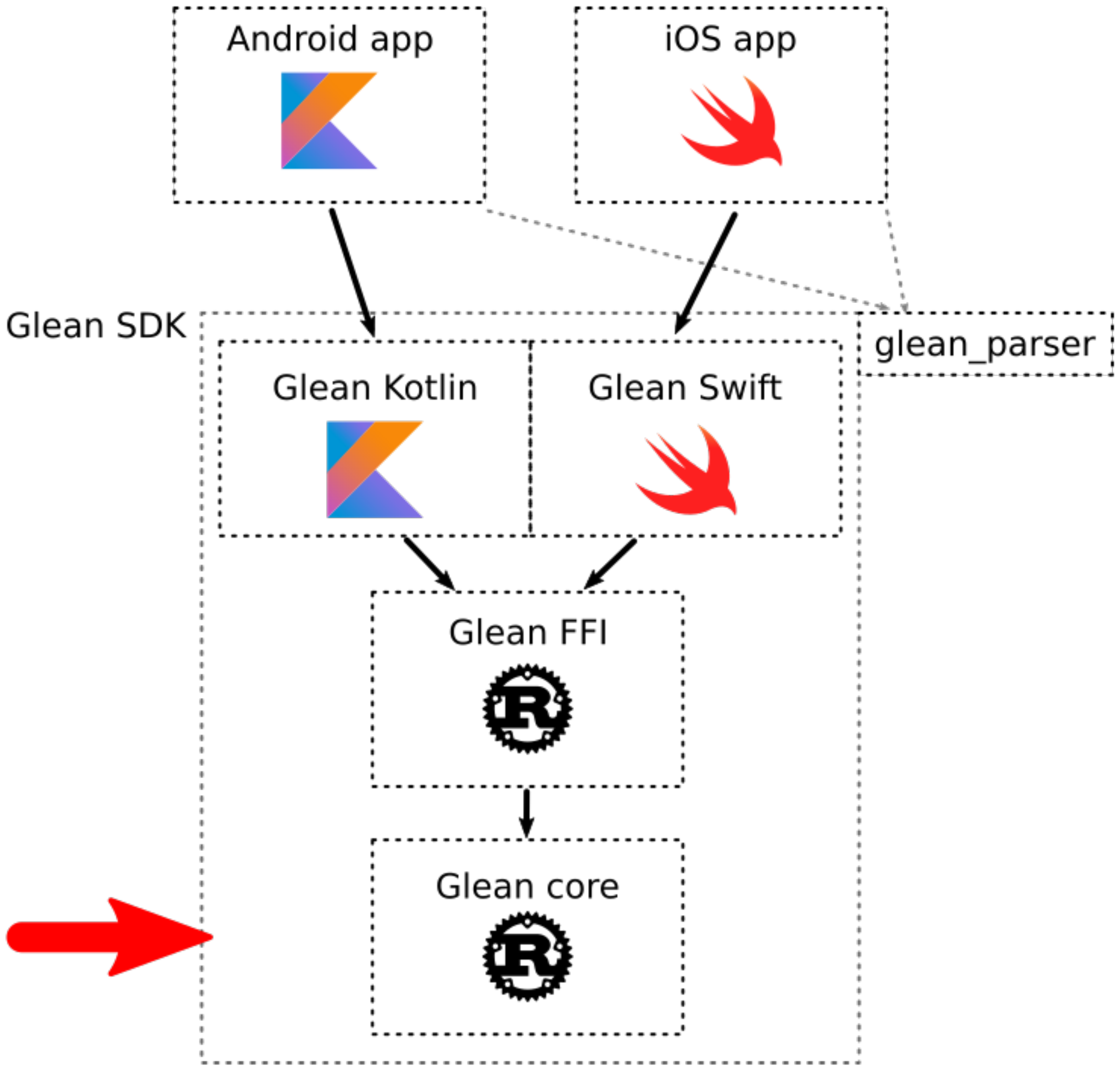
(and how no one noticed)

How Glean was built in the beginning



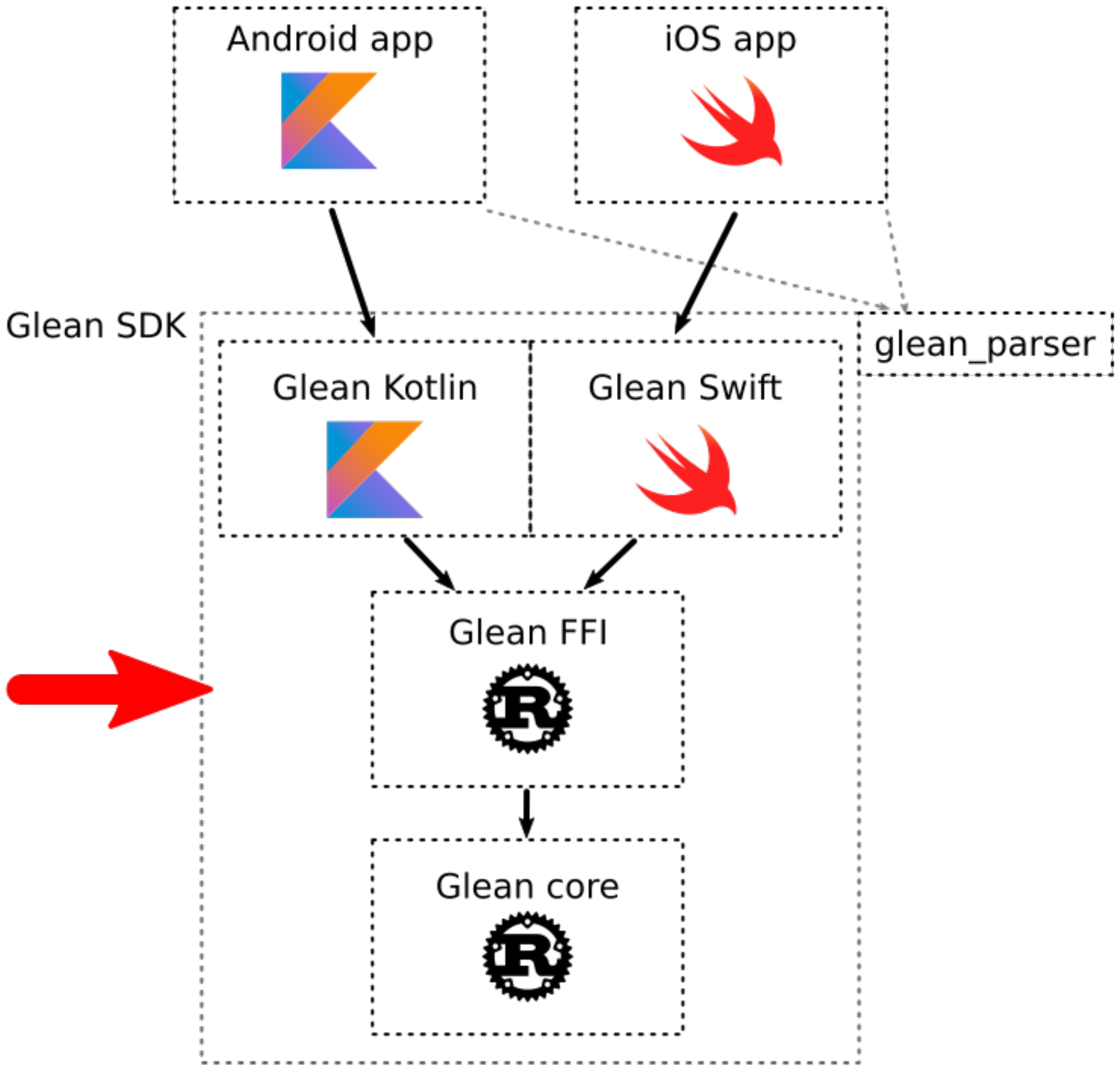
How Glean was built in the beginning

- Rust core implementation



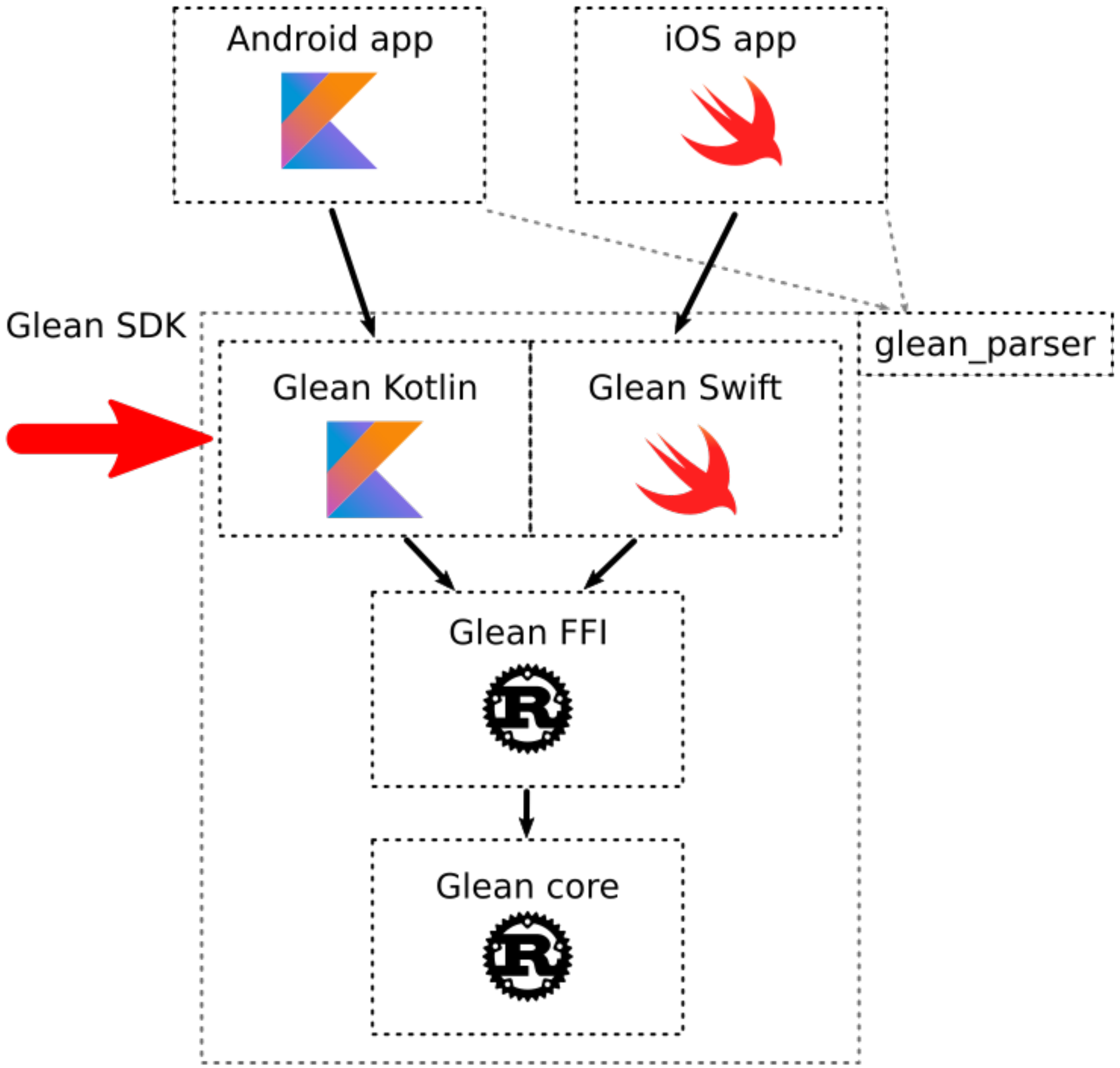
How Glean was built in the beginning

- Rust core implementation
- The FFI layer



How Glean was built in the beginning

- Rust core implementation
- The FFI layer
- Foreign language implementation



What if we don't need to write all that code?

A UniFFI is born

Defining the API once

```
// WebIDL-like interface definition language
interface Rocket {
    constructor(string name);
    void lock_steering(string direction);
    [Throws=LaunchError]
    boolean launch();
};
```

Implement it in Rust

```rust
struct Rocket {
  name: String,
  direction: String,
}

impl Rocket {
  fn new(name: String) -> Rocket {
    Rocket {
      name: name, steering: "".into()
    }
  }
```

```rust
fn lock_steering(&mut self, dir: String) {
  self.direction = dir;
}

fn launch(&self) -> Result<bool> {
  if self.direction != "up" {
    return Err(LaunchError::RocketLaunch);
  }
  Ok(true)
}
```

✨ 🪄 MAGIC! 🪄 ✨

Use it in Kotlin

```kotlin
import rediger.uniffi.rocketscience.*

val rocket = Rocket(name = "Orbiter")
rocket.lockSteering("up")
rocket.launch()
```

Use it in Swift

```swift
import rocketscience

let rocket = Rocket(name: "Orbiter")
rocket.lockSteering("up")
try! rocket.launch()
```

Use it in Python

```
from rocketscience import *

rocket = Rocket(name = "Orbiter")
rocket.lock_steering("up")
rocket.launch()
```

uniffi[7] - a multi-language bindings generator

- Side-project by Ryan Kelly in the summer of 2020
- Re-use known things
  - Auto-generate patterns used in appservices/Glean
  - WebIDL: Firefox already uses that

[7] github.com/mozilla/uniffi-rs

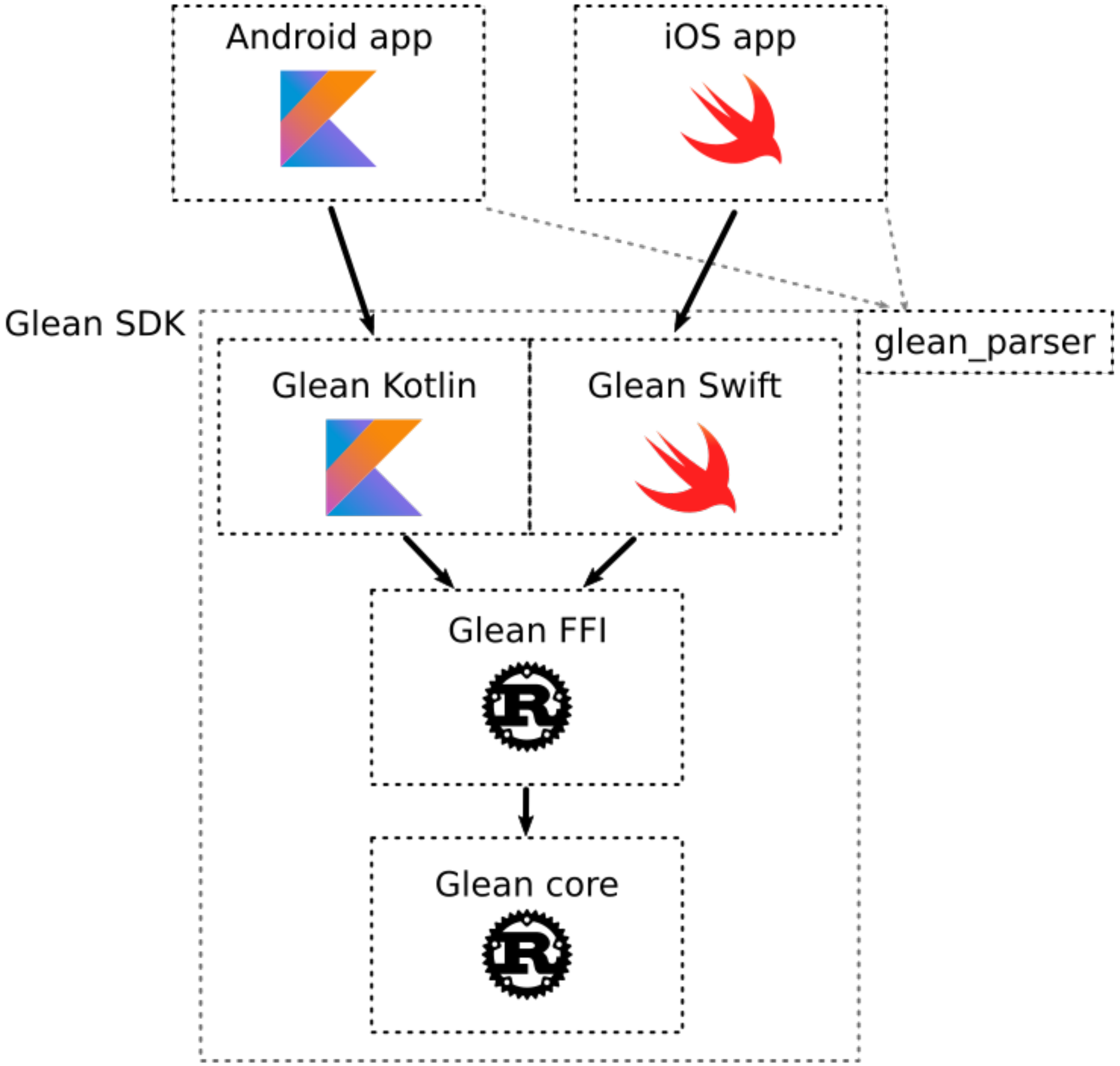
UniFFI is a success

- app-services migrated components one-by-one (mostly done)
- Frequent cross-team work and shared ownership between Glean & app-services

UniFFI is a success

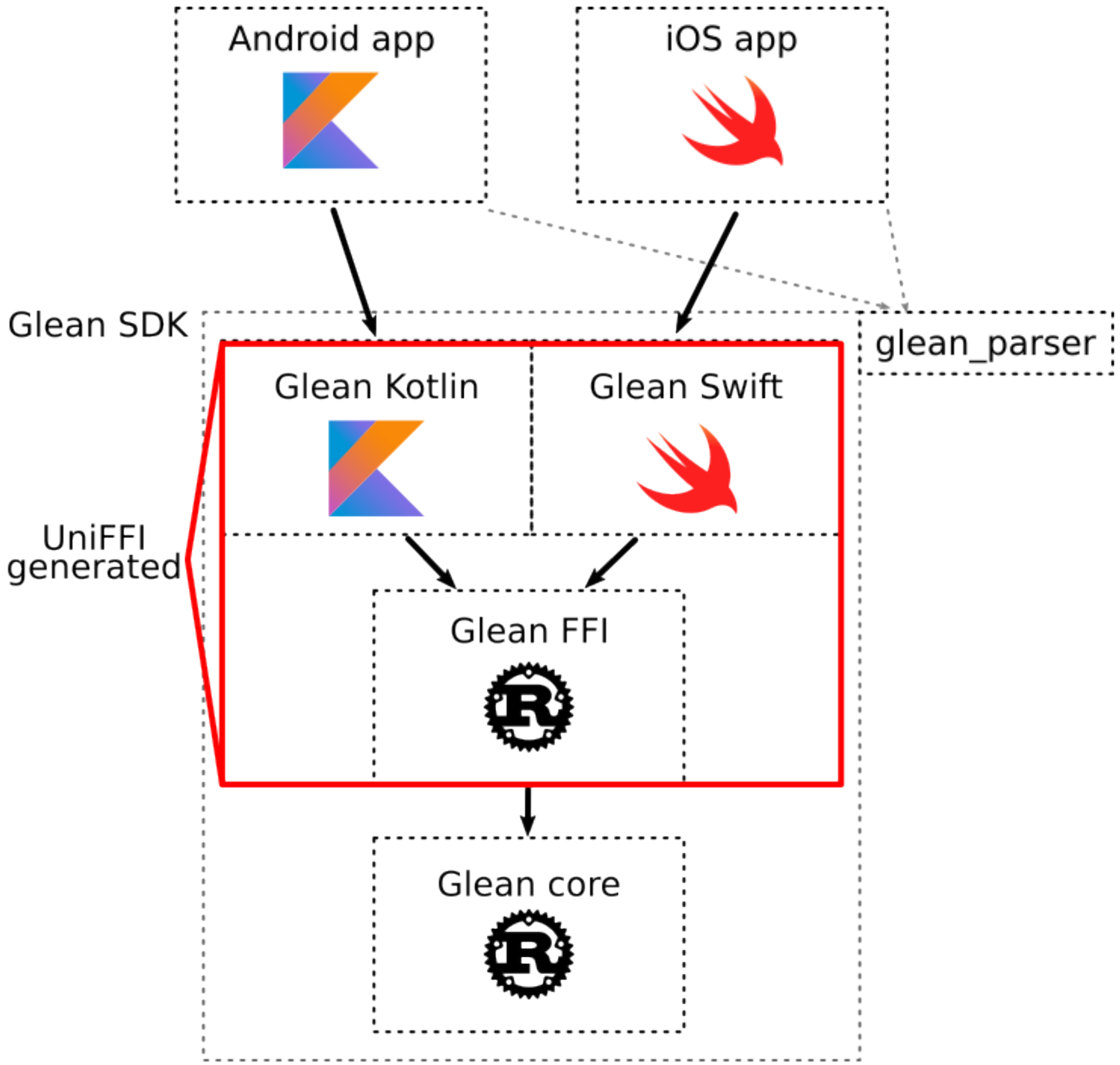
- Open-source projects with outside users & contributors
  - More language support (Ruby)
  - New features
  - Knowledge sharing with other tools (diplomat-rs)

How Glean is built now



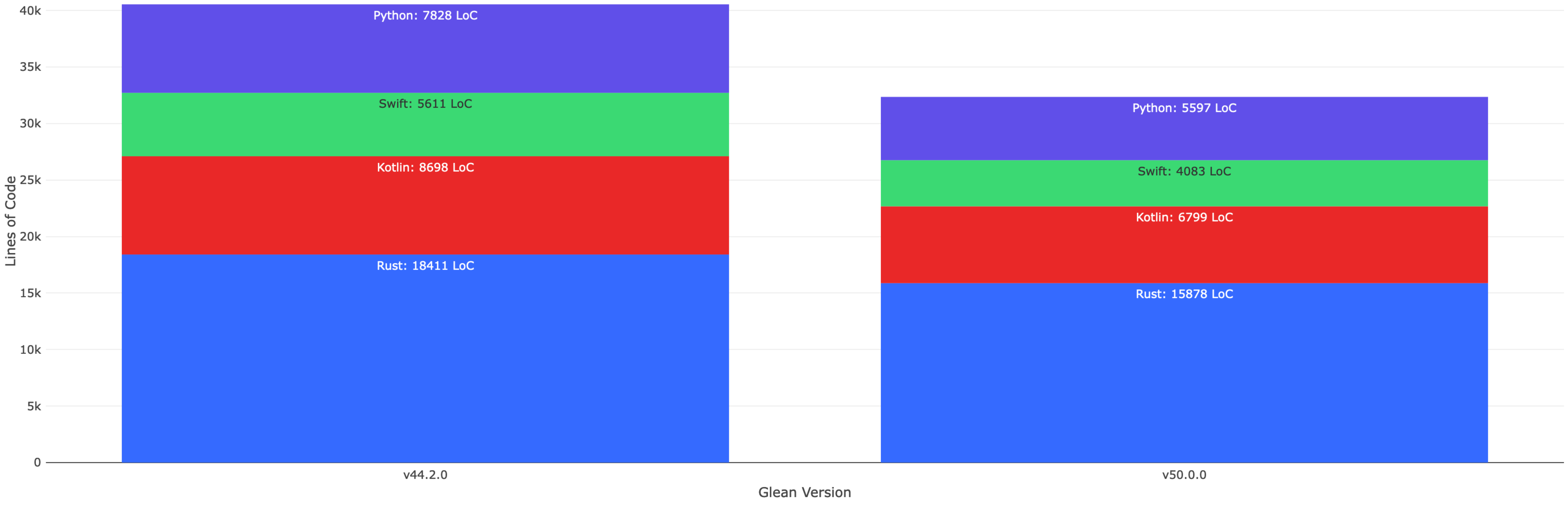
How Glean is built now

- Full API defined in `glean.udl`
- Full implementation in Rust
- UniFFI generates all bindings
- Minimal code in the target language to adjust API



What did that get us?

Less code!



What did that get us?

- Similar APIs across all target platforms without effort
- Simpler to add new metric types
- More consistent testing
- Simpler iOS megazord integration
- Trailblazed the way for UniFFI in mozilla-central

Glean v50 - the rollout

Glean v50 - development

- First plan in July 2021 (doc)
- Proposal accepted in October 2021 and development work started
- 2 people team (Thanks, Travis!)
- Bug fixes & feature development on UniFFI needed
- Often interrupted by other Glean & data work

Glean v50 - rollout planning

- Testing & Rollout plan in May 2022 (doc)
- Careful consideration of impact
  - Coordinated timing of landing
  - Schedule QA work ahead of time
  - Close work with affected teams before and during the rollout
  - Monitoring dashboard
  - go/no-go decision and rollback possibility

Glean v50 landed in Fenix Nightly v103 on June 9th

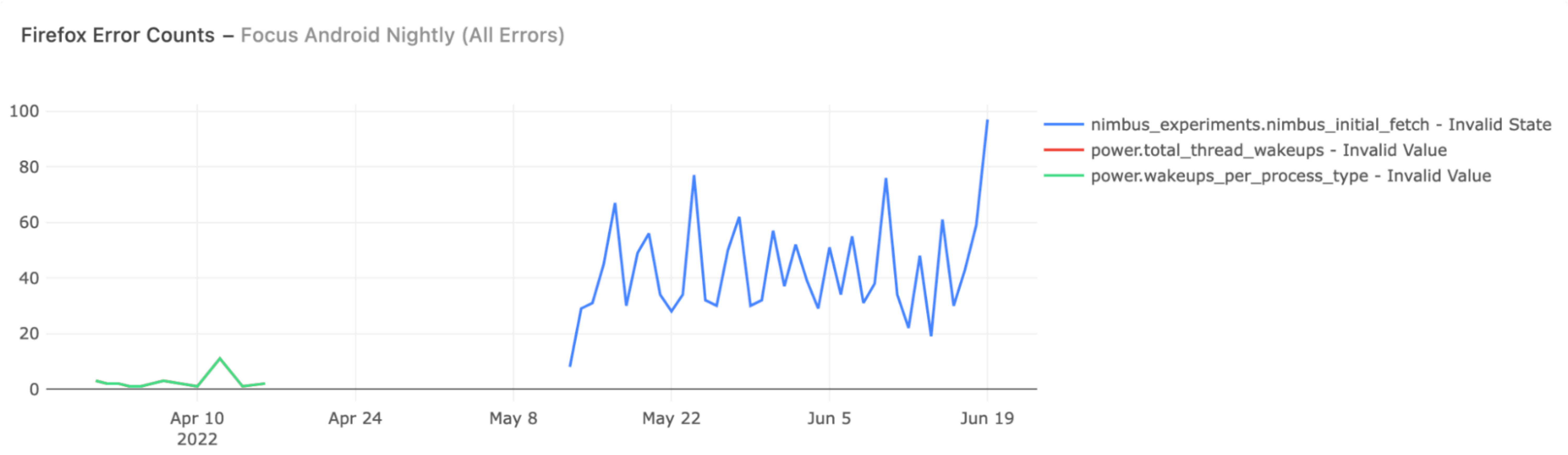
Glean v50 landed in Fenix Nightly v103 on June 9th

Did anyone notice?

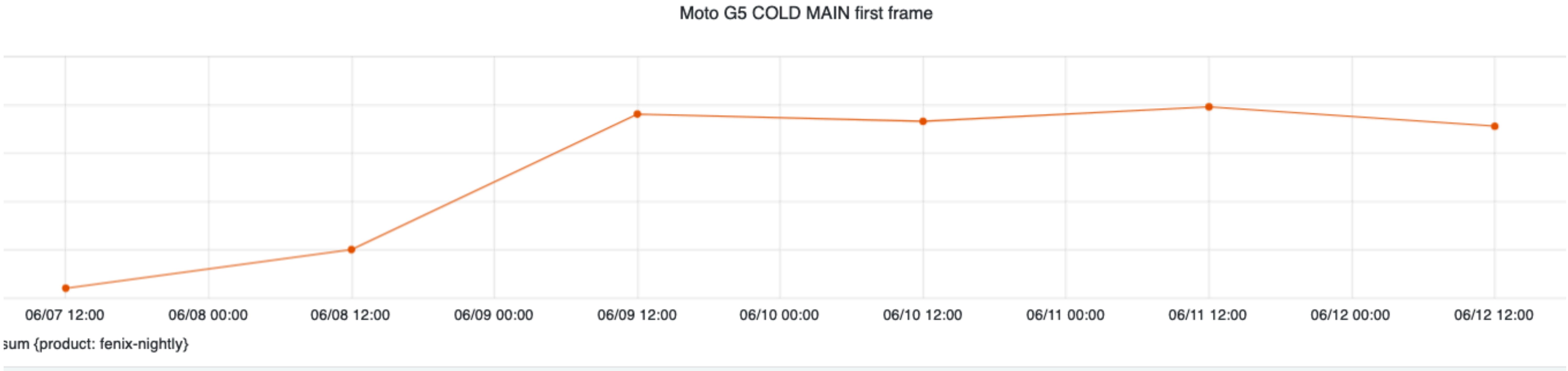
Glean v50 in Fenix v103

- QA finished on June 15th
- Data monitoring & analysis 1 week after rollout finished (doc)

Glean v50 - Error spikes for app metrics



Glean v50 - Startup regression



Glean v50 - The rest

- Minor bugs in the integration delayed initial merge
- Increase of intermittently failing tests - already fixed

Glean v50 - all good?

Glean v50 - all good?

No

Glean v50 - all good?

- July 19: Fenix v103 released
- July 24: Chris Peterson notices missing data in Fenix v103
- July 25: Glean team notices error increase

Glean v50 - all good?

- July 19: Fenix v103 released
- July 24: Chris Peterson notices missing data in Fenix v103
- July 25: Glean team notices error increase

We have a Data Incident!

Glean v50 - we have a data incident

- From Glean v50 on 4 Glean-provided metrics were not properly recorded
- July 25: Bug fixed in Glean
- July 29: Rolled out to Fenix v104 Beta
- August 3: Subsequently fixed in Fenix v103.2

Glean ~~v50~~ v51 - all good?

It is now.

What have we learned?

- ~~Always be migrating~~
- Plan early
- Involve stakeholders early
- Data uncovers problems, not always your own
- Nightly/Beta data is not as closely monitored as Release

Thanks to

- the Glean team: Alessio, Chris, Travis, Perry
- the Application Services team
- the UniFFI team
- the Mozilla Android team
- the Release team, especially RyanVM

Questions?