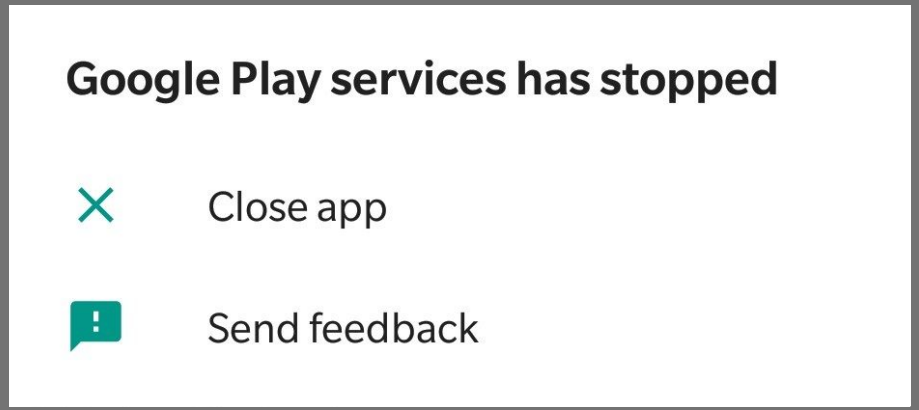
SRE for Mobile Applications

Samuel Littley
Mobile Infrastructure SRE, Google
samuellittley@google.com

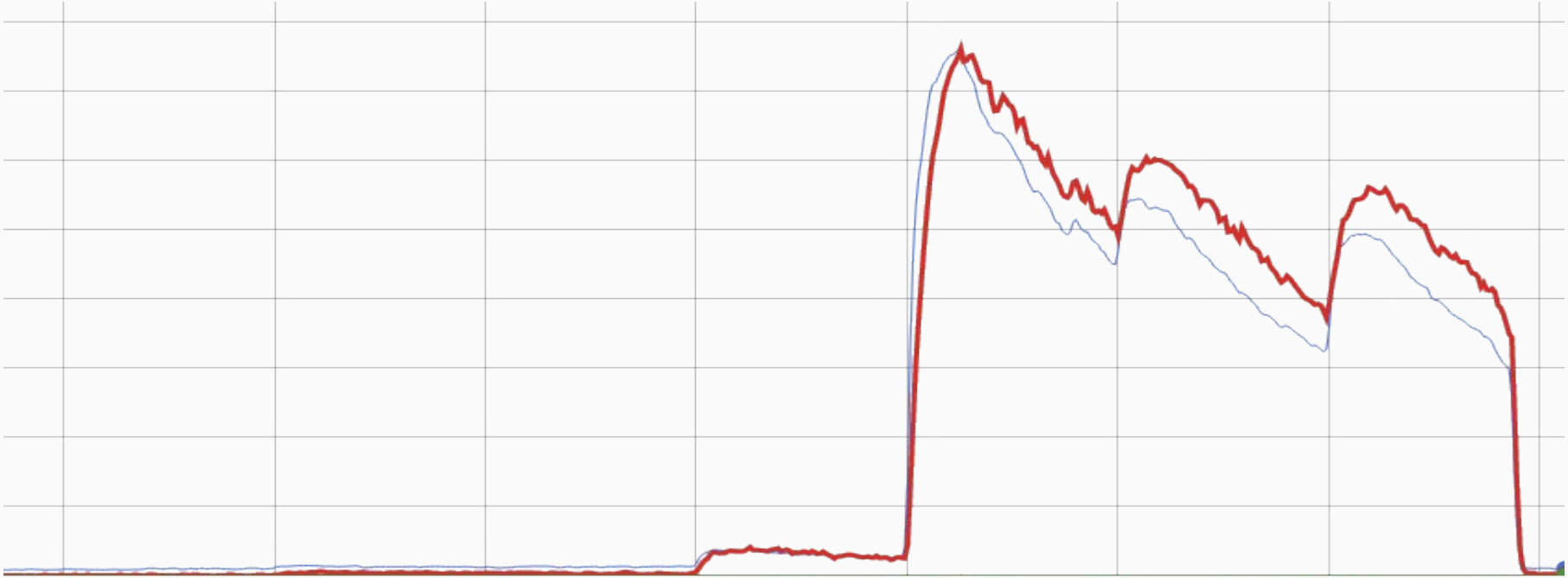

Google Play services has stopped
Close app
Send feedback

Agenda

- Why mobile?
- How to SRE a mobile app
- Case studies
- Future of SRE for Mobile

Why mobile?

- **Why mobile?**
- How to SRE a mobile app
- Case studies
- Future of SRE for Mobile



“In general, an SRE team
is responsible for availability, latency,
performance, efficiency,
change management, monitoring,
emergency response, and capacity planning.”

Ben Treynor, VP of Engineering, Google

“In general, an SRE team for mobile is responsible for availability, latency, performance, efficiency, change management, monitoring, and emergency response.”

Kristine Chen, Mobile Infrastructure SRE, Google

Advantages of client-side SRE

- Client-side experience is the user experience
- Richer metrics
- Potential for client-side improvements
- Powerful control options

Disadvantages of client-side SRE

- No direct control
- No rollbacks
- Scale
- Heterogeneous environment
- Imperfect monitoring

Why mobile? - Key Points

- Get the full end-to-end picture
- Provide a better experience for users

How to SRE a mobile app

- Why mobile?
- **How to SRE a mobile app**
- Case studies
- Future of SRE for Mobile

Monitoring

Realtime Monitoring

- Client calls logging API
- API sends logs to server
- Server reports metrics on behalf of clients

Caveats of realtime monitoring

- Cardinality
- Cost
- Reliability

Continuous Integration

- Emulators
- UI Tests
- Production Servers

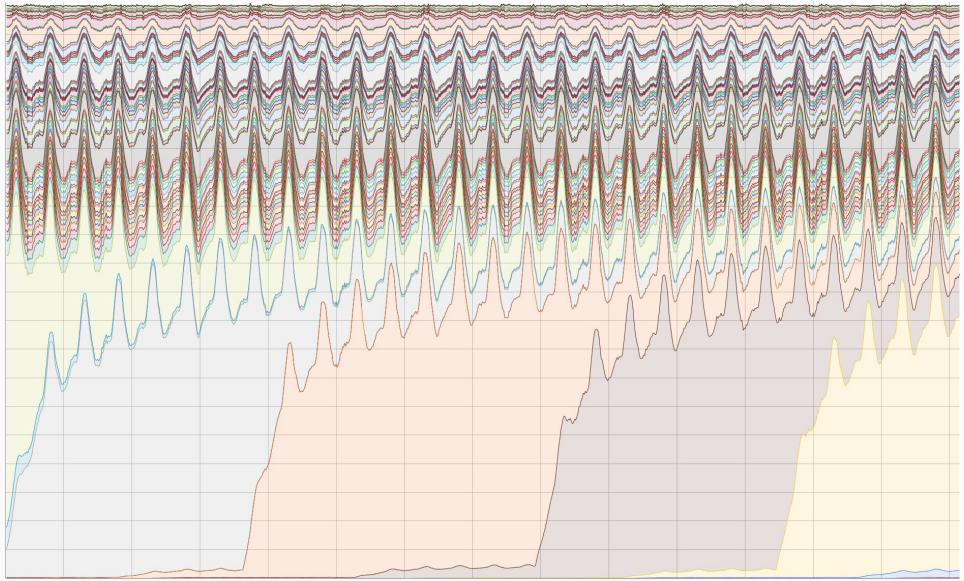
System and app health

- Battery
- CPU
- Memory
- Disk
- Network
- Jank

SLIs and SLOs

Client-side SLIs and SLOs

- High rate of client-side failure
  - Loose objectives
- Error budget hard to interpret
  - Small regressions lost in the noise
- Can only improve newer versions
  - SLI skewed by the long tail



Using SLIs/SLOs and Metrics

- Consider filtering unimportant failures
  - Beware blindness to potential improvements
- Manage via relative effect
  - Allow small regressions when necessary
  - Continuously repair

No-op binary releases & Progressive rollouts

Flag guarding

- Flag guard everything
- Boolean flags for new/changed code blocks
- Strings, integers, floats for controlling behaviour
- Rollouts randomised per flag to avoid interference
    - hash(seed + user_id) % 100 → percentage buckets

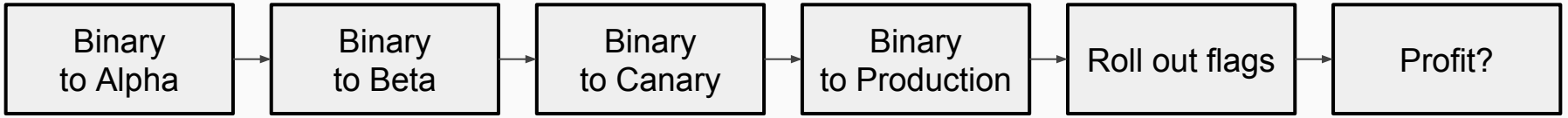
Caveats

- Not always possible
- Cardinality
- Attribution
- Binary release notes
- Feature release deadlines & public announcements

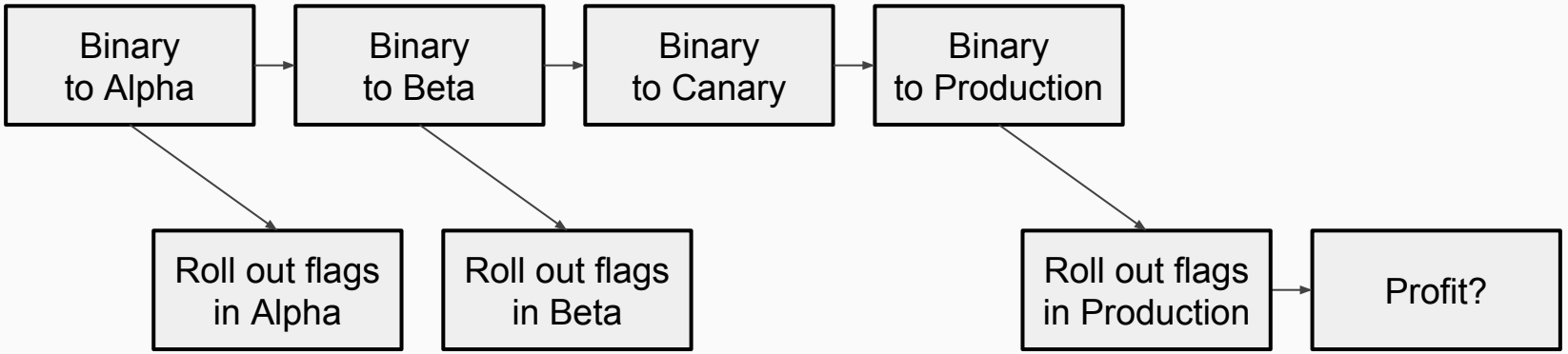
The big advantage

- Flags = control surface
- Primary mitigation step

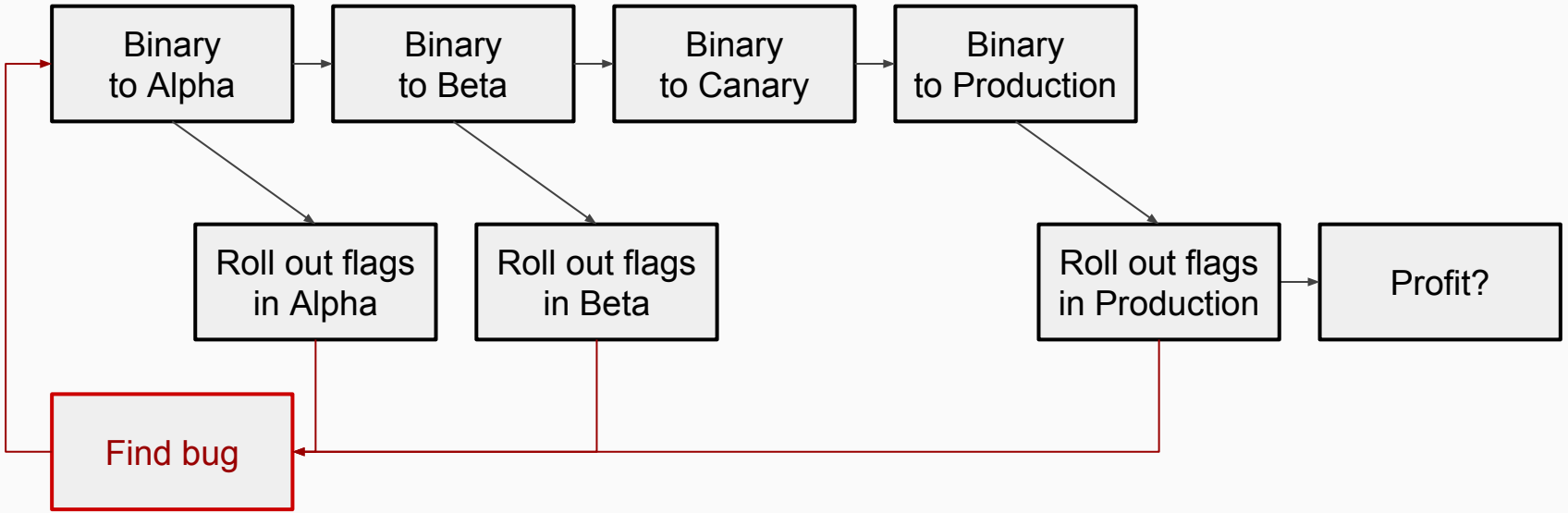
Release Process



Release Process



Release Process



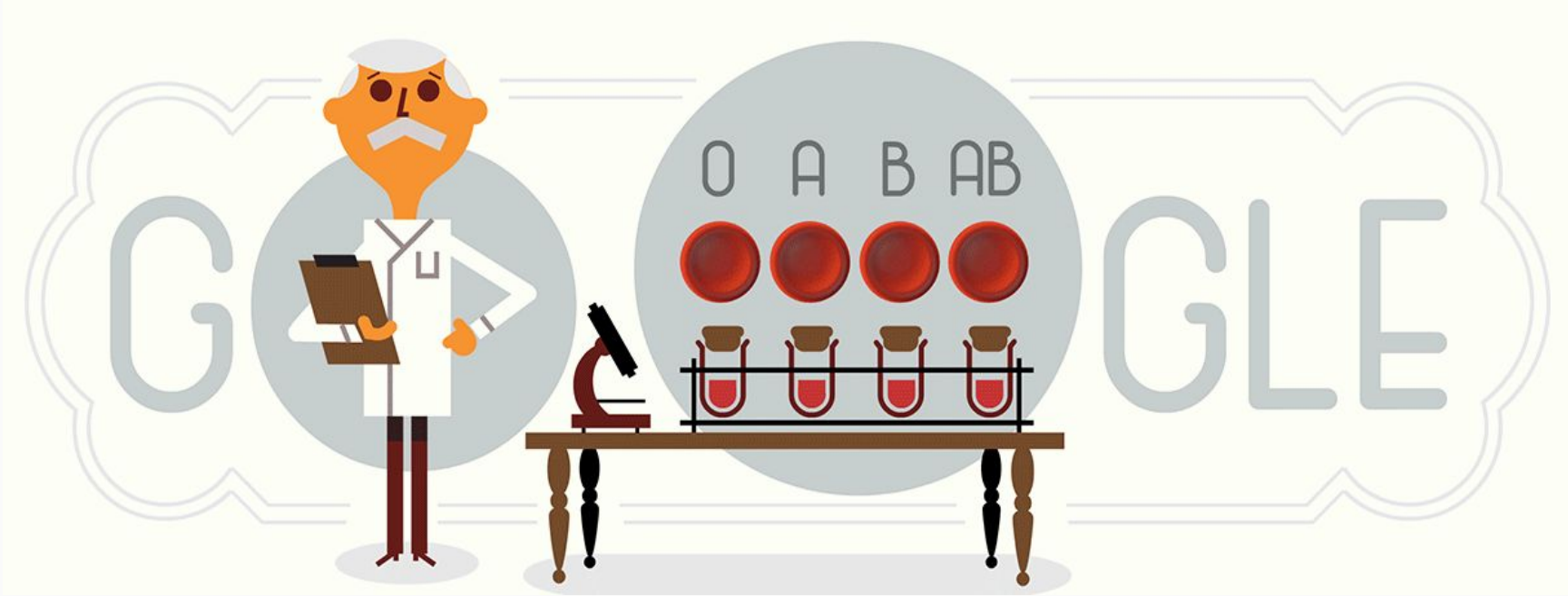
How to SRE a mobile app - Key Points

- Monitor from the client side
- Focus on release safety
- Flag guard as much as possible

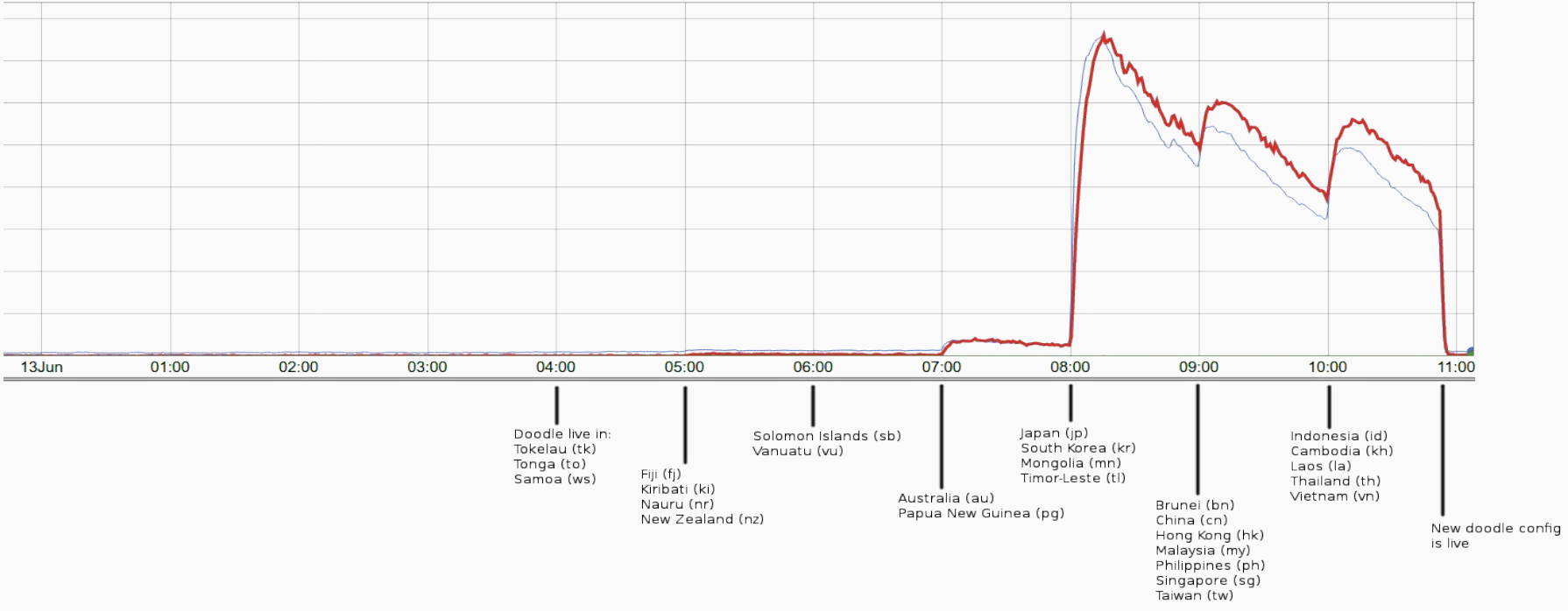
Case Studies

- Why mobile?
- How to SRE a mobile app
- **Case studies**
- Future of SRE for Mobile

Karl Landsteiner



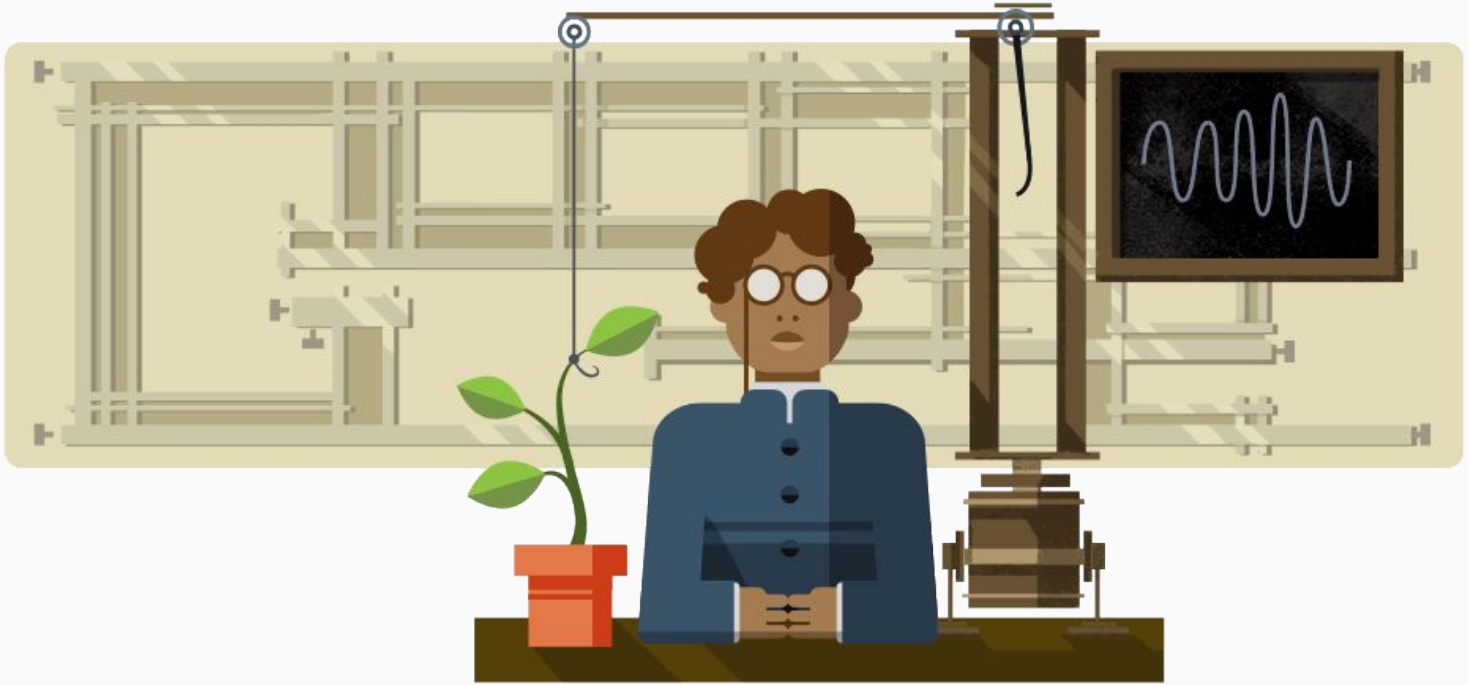
Doodle Crashes



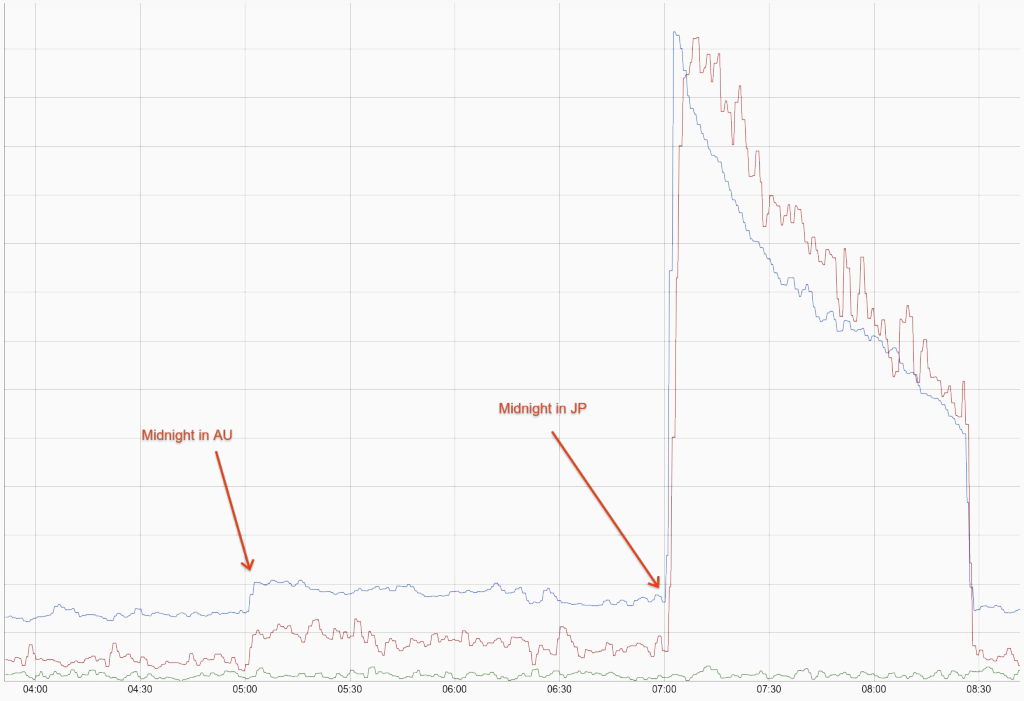
What did we learn?

- Mobile SRE
  - Know your configuration mechanism
- General SRE
  - Validate your inputs
  - Degrade gracefully

Jagadish Chandra Bose



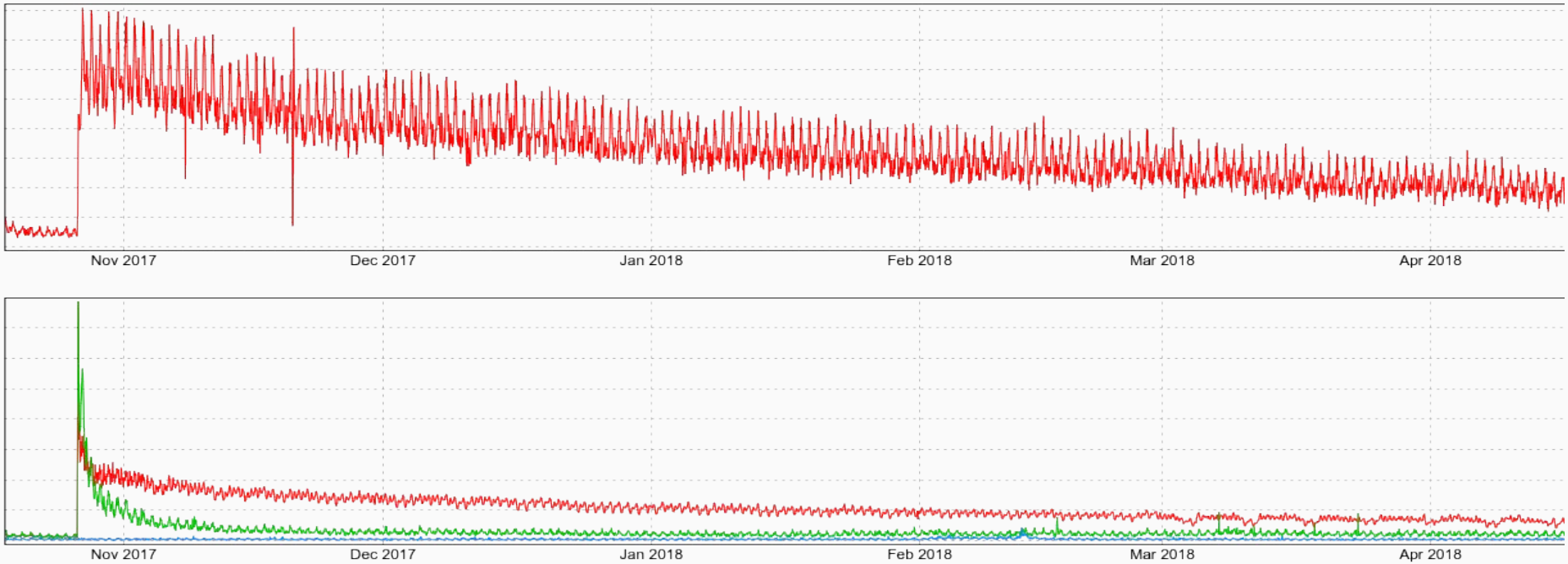
Doodle crashes (again)



What did we learn?

- Mobile SRE principles
  - Client-side fixes aren't sufficient
  - Logging is time-sensitive
- General SRE principles
  - Experience counts
  - Follow up on action items

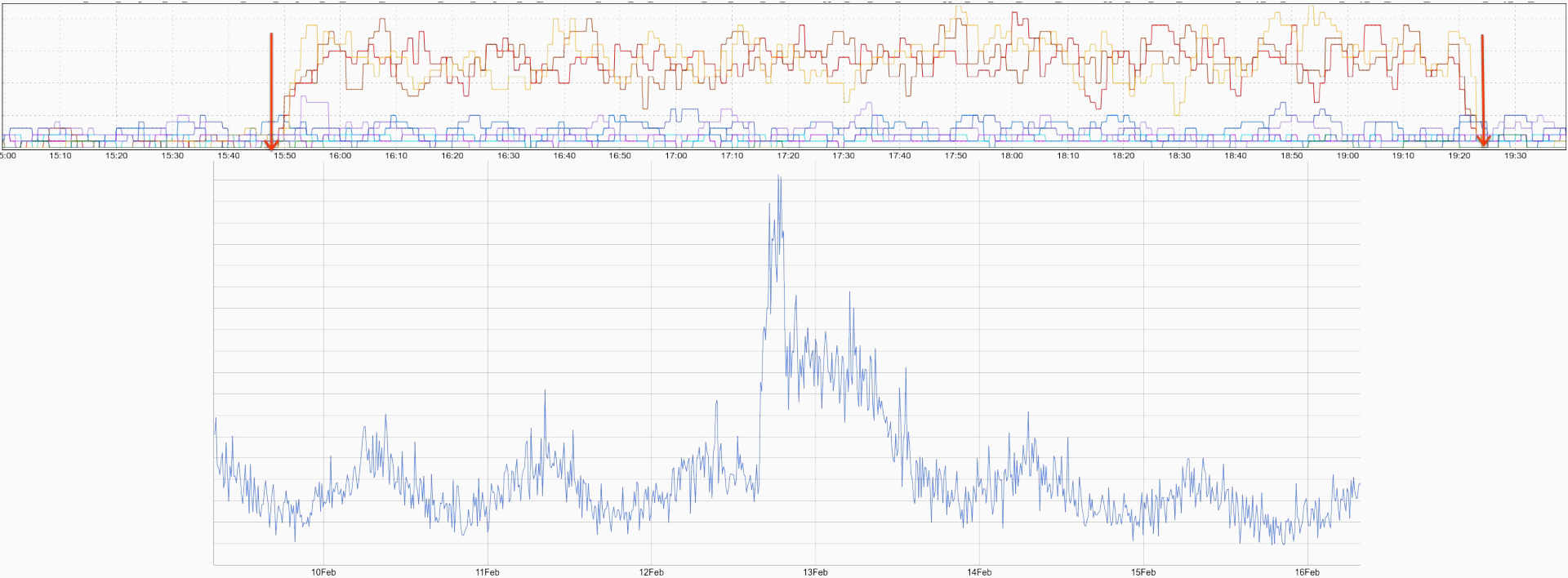
Cached config crashes



What did we learn?

- Mobile SRE principles
  - Caching config is risky
  - Quick mitigation is important
  - Sometimes, the effort is not worth the reward
  - Old versions can't be forgotten
- General SRE principles
  - Canary everything

Unresponsive app caused by fixes



What did we learn?

- General SRE principles
  - Fixes can cause more problems
  - Slow rollouts only work if the code is exercised
  - Root cause != trigger

Other Outages

- Play services update breaking Google apps
- Releases blocked on metrics (x2)
- Binary pushes to production without canary (x1.5)

What did we learn?

- Mobile SRE principles
  - User feedback is important
  - Sometimes, abandoning the release is OK
- General SRE principles
  - Automation > humans
  - Communication and organization skills matter

Case Studies - Key Points

- Be aware of differences between client and server
- Defense in depth
- Be cautious

Future of SRE for Mobile

- Why mobile?
- How to SRE a mobile app
- Case studies
- **Future of SRE for Mobile**

Consistency

- Monitoring, Feedback and Testing
  - Reliable
  - Low latency/short feedback loop
- Release and Experimentation
  - Standardised processes and tools
- Control Surface

In the near future

- Improved binary releases
  - Additional testing tracks
- Dynamic apps
- System/platform health

Unsolved problems

- Monitoring at high cardinality
- Handling of not-our-fault failures

Key Point?

Flag guard all the things!