

# BoF - iOS session

12/12/19

## Summary:

### Topic discussed

- Interest in RBE for iOS development
  - RBE on macOS is quite complicated, hard to manage Macs + Xcode versions
  - Lot of interest in macOS RBE support
- Rules\_apple/rules\_swift new release policy: internal forks of these repositories won't scale well, managing version compatibility yourself is difficult
  - Federation should help improve this but the cost of maintaining a fork will still be present
- Communication on open issues, PRs, and priorities
- Improving migration process from Xcode → Bazel
  - Support for headermaps (Bazel by default requires imports to be relative to the workspace root)
  - Support for mixed Swift-ObjC modules
- Code coverage with Bazel
- Xcodeproj generation with Bazel

### Specific Requests and Action Items

- Better documentation regarding (iOS) RBE: it's difficult to set it all up and easy to get wrong
- Community driven roadmap: expanding functionality based on community needs
- Improve communication on open Bazel and rules\_apple/rules\_swift issues; be more transparent with priorities and timelines; make sure all untriaged issues are assigned a priority
- Get code coverage working for iOS tests
- Better support for community contributions; accept pull requests or support a shared fork
- Add a provider for headermaps (e.g. via objc\_provider?)
- Make it easier for external contributors to add rules/projects to the federation
- Make it easier to extend Bazel and rules; add override hooks and extensibility features so that external contributors don't need to fork code
- Make the initial setup easier; make it easier to build an Xcode/iOS app with Bazel
- Make a Bazel community GitHub org where we give community members more privilege