# **SWG 2022-02-21 Agenda**

* Update on the GHC.X.hackage proposal
* Relationship with HFTT
* Update on GHC tick-tock releases
* What can we learn from Aeson 2.0 (followup)
* Discuss [Cabal and Cabal-syntax split](https://github.com/haskell/cabal/pull/7620#issuecomment-914180574,%20%20%20%20%20https://github.com/haskell/cabal/issues/7974)
* Maintainership standards <https://github.com/haskellfoundation/stability/issues/7>

## Notes

* Simon: Should our discussions go through the tech track process?
  + Overall consensus: avoid re-inventing the wheel; let's use the TT process as our mechanism for consulting the community.
  + Let’s try it in the case of GHC.X.Hackage
* GHC.X.Hackage:
  + No motion yet, will try to get to this in the next week
  + Next step: **Ben**: open an MR against tech-track
* GHC tick-tock releases
  + **Chris & Ben**: work on a proposal (in a google doc)
* Aeson-2.0 (continuing from last week):
  + Last week it was suggested that aeson-1.0 should have started the interface changes to allow smoother upgrade path
  + Continuing now as Adam, an aeson maintainer, is now here
  + SPJ question: what is our role here? What is the problem we are trying to solve? Are we developing a proposal? Or what?
  + Answer: expected outcome is a short "war-story" report that we can share with the community, making suggestions for the future that would help ameliorate the costs of change. **Chris** will draft this. Adam: maybe we can make it less Aeson-specific, less like accusing a hard-working colleague. E.g. articulate the "general principle" in the cabal item next -- or "Guidelines for package maintainers".
* Cabal and cabal-syntax split
  + <https://github.com/haskell/cabal/pull/7620> , <https://github.com/haskell/cabal/issues/7974>
  + Parser for .cabal files is in a new package, cabal-syntax
  + Package cabal depends on package cabal-syntax
  + But cabal does not re-export cabal-syntax; hence presumably a major version bump (since it slims down the cabal API)
  + So some packages that currently depend on cabal will now need to depend on {cabal, cabal-syntax}; others can depend only on cabal (a smaller and more stable API).
  + Tom argues: make a meta-package that unions {cabal-syntax, cabal-core} that keeps the same API, but encourage people to depend only on cabal-core if that's all they need. (There is a payoff for doing so, at a time of your choosing, because it means you depend on a smaller and stabler API.)
  + General principle: provide opportunities for more refined dependencies, but try not to use API breakage to force people to take action at a time they may not choose.
  + Action  **Mikolaj** to take back these suggestions to the Cabal team, and (if they wish) initiate a dialogue. Ben has posted on [the issue tracker here.](https://github.com/haskell/cabal/issues/7974#issuecomment-1047057577)
* Richard's suggestion <https://github.com/haskellfoundation/stability/issues/7>
  + Opt-in = good. We can judge through uptake.
  + What is the incentive to join in? One answer: get Hackage to sort/filter based on badge. (A form of SEO.)
  + A concern: varying opinions about some metrics, e.g. testsuite coverage.
  + Syd: problem is not too many packages, but rather problems with packages themselves.
  + Tom: good idea in principle. Details in Richard's proposal are pretty tentative.
  + Syd: objective metrics are helpful, not an opaque authority handing out gold stars. Subjective measures (e.g. upvotes) in Hackage have not worked.
  + Rate and recency of commits is a relevant metric. [Albeit fakeable.]
  + We should involve Hackage maintainers/trustees before going to a HF TT proposal.
  + Who would like to lead? We could ask Richard, but he is probably over-subscribed. If so, **Adam** will drive, consulting Richard.