# Node.js Technical Steering Committee (TSC) Meeting 2021-04-08

## Links

* **Recording**:
* **GitHub Issue**: $GITHUB_ISSUE$
* **Minutes Google Doc**: $MINUTES_DOC$

## Present

* Anatoli Papirovski @apapirovski (TSC)
X* Beth Griggs @BethGriggs (TSC)
X* Ruben Bridgewater @BridgeAR (TSC)
X* Сковорода Никита Андреевич @ChALkeR (TSC)
* Colin Ihrig @cjihrig (TSC)
* Shelley Vohr @codebytere (TSC)
* Daniel Bevenius @danbev (TSC)
* Franziska Hinkelmann @fhinkel (TSC)
* Gabriel Schulhof @gabrielschulhof (TSC)
X* Gireesh Punathil @gireeshpunathil (TSC)
* James Snell @jasnell (TSC)
* Joyee Cheung @joyeecheung (TSC)
* Mary Marchini @mmarchini (TSC)
X* Matteo Collina @mcollina (TSC)
X* Michael Dawson @mhdawson (TSC)
* Myles Borins @MylesBorins (TSC)
* Michaël Zasso @targos (TSC)
X* Tobias Nießen @tniessen (TSC)
X* Rich Trott @Trott (TSC)

X* Robert Naggy
X* Danielle Adams

## Agenda

### Announcements

* Security updates were released on Tuesday for all active release lines.
* Node.js 16.0.0 is coming. Last release candidate is expected to be next week.
* Node.js 10 will go end-of-life at the end of April.
* Node.js 15 will go end-of-life in June.

### CPC and Board Meeting Updates

*Extracted from **tsc-agenda** labeled issues and pull requests from the **nodejs org** prior to the meeting.

* Pending charter changes in the TSC repo in a pull request.
* Mary is our rep again for the next year. Congratulations, Mary!

### nodejs/node

* doc: mark callback-based fs API as legacy
[#37948](https://github.com/nodejs/node/pull/37948)

* Matteo: Last comment by Tobias matches my sentiment.
* Robert: Callback API is unsafe. If you use a file descriptor after closing, behavior is undefined. This problem does not exist with promise API.
* Gireesh: Does that qualify for making it Legacy?
* Robert: It’s a motivation.
* Gireesh: Legacy status landed recently. Should we have more APIs Legacy before moving this one?
* Michael: That’s my feeling. What do we do to move it forward?
* Rich: What’s a good timeline/ordering of things to get it to Legacy?
* Matteo: I don’t see a good path. Callbacks are very deeply entrenched. The major ones missing are all the networking parts. This is not a small amount of work. We’re working on it. Then there is the thing around Node.js streams which would involve pulling the callbacks out of C++ somehow.
* Rich: Maybe we just leave it alone.
* Robert: Do we have a way to discourage usage other than legacy/deprecation?
* Nikita: Perhaps we can emphasize the promise APIs in the docs.
* Tobias: We’ve done that in the past.
* Michael: +1
* Robert: We need to encourage usage of the Promise-based APIs.
* Michael: Docs are a good first step.
* Tobias: Ecosystem adoption of promises in 3rd party packages is important. Many packages still don’t support promises themselves, and mixing them with promises is troublesome.
* Michael: Leave it on the agenda for next week. Let’s see if we can get a champion or champions to take this on and figure it out.)

* http: significant performance regression on master
[#37937](https://github.com/nodejs/node/issues/37937)
  * mostly resolved, so no discussion needed.
  * If you see code that might affect perf or http, (event emitter, ..) please run performance regression

* Robert: Pre-release task?
* Matteo: There’s already a lot on the releasers checklists.
* Michael: People need to run the benchmarks when they see code that might be affected.
* Ruben: Benchmarks need to be revised. It takes weeks to run them. We can’t have the releasers do that.
* Robert: Maybe streams and http teams run the relevant benchmarks. We were lucky Matteo is doing the benchmarks for streams etc.
* Ruben: It’s often primordials that cause the performance regressions. It’s not obvious that primordials would have a performance decrease. It has happened so often that I fear we might have reduced our perf in multiple cases without noticing.
* Matteo: I agree. People with knowledge of subsystems have not been tagged into primordial PRs.
* Ruben: It’s not only streams. Lots of other performance sensitive areas. util.inspect is an example. I do not see any benefit to using primordials. You will have lots of external code vulnerable.
* Michael: Sounds like a separate issue.
* Ruben: I’ll open an issue about that.
* Michael: Maybe we need a higher bar for primordials changes.

* lib: refactor Timeout, ImmediateList, TimersList to class [#37408](https://github.com/nodejs/node/pull/37408)
  * On agenda for visibility, going to remove from agenda

* deps: add Yarn 1.22.5 [#37277](https://github.com/nodejs/node/pull/37277)
* Danielle: As a user, I like it. As a platform maintainer, I’m less supportive. There’s already a package manager bundled, and I get comments all the time from Heroku users about removing stuff, so bloat is an issue.
* Tobias: Ideally, we’d have everything separated and then some convenience distributions. I don’t have an opinion. Might be better for some and not at all better for others. Always the option to build from scratch though. So, sure, why not include it, I guess, as long as it’s not extra work for maintainers.
* Michael: npm is a source of security updates so might yarn cause even more security releases? `npm install yarn` doesn’t seem like a huge extra step.
* Beth: My main thought is it doesn’t seem fair to hold it up on a survey. That seems like a lot of hoops we’re asking the contributors to go through to have their PR considered. Because of that, I’d be +1 for it to go to a vote so it leads to a resolution.
* Gireesh: Tradeoff between fairness and maintenance burden?
* Michael: Yes, that’s the key issue.
* Gireesh: On the one side, we want to be fair. On the other hand, it implies complexities for maintenance. So the question is what do we get for the community? This sort of thing will come up again in the future. What exactly are the community’s priorities for Node.js? Do we want to be seen as fair? Stable?
* Michael: Once you include a second one, where does it stop? Potentially slippery slope. So

“fairness” may not be the way we want to think about it. Benefit to the end user is the way to think about it, perhaps.
* Ruben: What happened to the proposal to remove npm and supplying a tool that allows people to install whatever package manager they want? I would support that. Convenience for users is the top priority from my perspective. They could install whatever they like. We don’t have to maintain a lot of dependencies. We would still have decisions about what package managers to include.
* Michael: If we don’t have enough information right now, I’m comfortable saying, “Not never, but no for now.” Unless someone is volunteering to lead the survey effort, or if Rich can get an RFC going next week, then OK. But given those things don’t seem likely, we should perhaps vote on this PR.
* Matteo: What happened to corepack? Why don’t we land that? I would prefer to ship this to shipping yarn.
* Beth: Does it make sense for the TSC to vote on both PRs?
* Michael: That would make sense to me. We could see if we can do it in a way that people can say “yes” to both but that they prefer one over the other, for example.
* Robert: If fairness is the core issue, removing npm is better than adding more package managers.
* Gireesh: Are security issues delegated to the package manager maintainers in that case?
* Matteo: corepack literally has no runtime dependencies.
* Michael: If yarn is installed transparently via corepack, does it look like we are responsible for security issues in it?
* Nikita: I’m against bundling yarn directly but I’m in favor of a script to install a package manager. I’ll elaborate on GitHub later.
Action: Michael will open an issue in the TSC repo to gather comments before moving to a vote. Basically, pick corepack, pick bundling yarn, or pick neither.

* TLS: improve compliance with shutdown standard, remove hacks [#36111](https://github.com/nodejs/node/pull/36111)

* Rename default branch from "master" to "main" or something similar [#33864](https://github.com/nodejs/node/issues/33864)

### nodejs/TSC

* Revist the meeting times [#1004](https://github.com/nodejs/TSC/issues/1004)

* Nominations for TSC Membership [#1000](https://github.com/nodejs/TSC/issues/1000)

* any interest in an RFC process? [#962](https://github.com/nodejs/TSC/issues/962)

* Apple Silicon plan [#886](https://github.com/nodejs/TSC/issues/886)

### nodejs/security-wg

* Potential stagnation of open issues on h1 bounty program [#654](https://github.com/nodejs/security-wg/issues/654)

### nodejs/undici

* Added WG Charter [#626](https://github.com/nodejs/undici/pull/626)

## Strategic Initiatives

## Upcoming Meetings

* **Node.js Foundation Calendar**: <https://nodejs.org/calendar>

Click `+GoogleCalendar` at the bottom right to add to your own Google calendar.