The Android 11 interview: Googlers answer our burning questions



Aurich Lawson / Getty Images

reader comments

60 with 43 posters participating

Share this story

We've established a bit of a tradition here at Ars. Every year at Google I/O, we have a sit-down talk to learn more about Android directly from the people that make it. Of course, this year, just about every major event was canceled due to the coronavirus pandemic, nothing is really normal, and Google I/O never happened.

We can still do interviews over the Internet though! So while it happened later in the year than normal, we were still able

to hold our annual chat with some of the most important Googlers at Android HQ: Dave Burke, Android's VP of Engineering, and Iliyan Malchev, Principal Engineer at Android and the lead of Project Treble.

We came prepped with questions about the more mysterious corners of Android 11, which actually led to a lot of interesting talk about the future. You'll learn about a coming re-write of the Bluetooth stack, and there's lots of talk about modularity and easy updating (like plans will hopefully, someday, allow you to update the Linux kernel and developer APIs as easily as you download an app update).

The Android Interview Series

View more stories

This interview took place just days before the launch of Android 11, which went final on September 8t. As usual, this has only been lightly edited for clarity, and I'll include whatever background is needed in italics. Given the odd state of everything as we all popped into a Google Meet video chat, COVID was an obvious first topic.

Ars: How are you all dealing with COVID-19 in Android development land?

Dave Burke: Good. Like everyone, when we did the work-from-home switch, it was a bit of a scramble to say the least. We had a lot of developer productivity things to iron out. A lot of people use high-powered workstations for building the operating system, with a phone tethered over USB, and we wanted to find a way that they could still use their workstations but flash to a phone that was tethered to a laptop. So we did a bunch of infrastructure work and whatnot to get everyone up and running. That actually worked pretty well. I was pretty impressed.

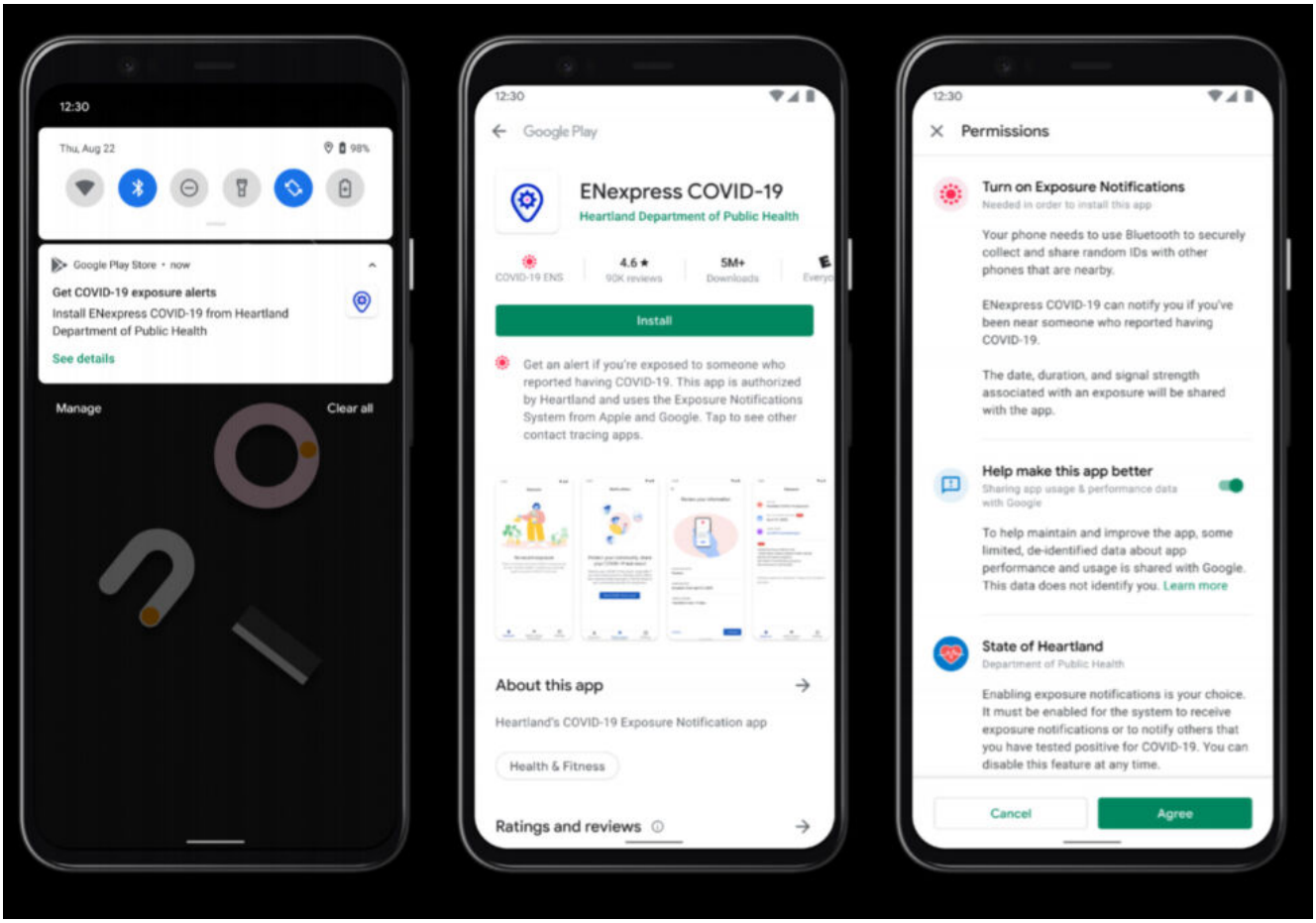
Google Meet worked out great, too. I remember thinking a couple years ago, “Wow, Google is investing a lot in this video conferencing stuff, why not use something commercial?” But now I’m so glad Google built its own. A lot of us now use the Meet rooms, so we have a lot of like virtual, Slack-like channels, if you want to call it that. It’s been pretty good!

I mean it’s obviously been tough as well, not having things like corridor conversations. We saw somewhere between like a 7 to 11 percent drop in productivity, I would say, at the beginning. Then it seemed to recover as people adapted. We did set the schedule back about a month just because we need to accommodate folks with that transition. The industry needed time to adapt, too, but yeah, it’s mostly working out. Of course, I think we would all be quite happy to go back to work.

Iliyan Malchev: I think the biggest worry going into COVID was just, ‘Will the ISP infrastructure be able to handle this huge spike in media consumption?’ It seems to have held up for the most part.

Burke: The other thing I’ve been working on is is exposure notifications with Apple, so that’s been pretty intense. We’re building capacity to detect if you’ve been approximate to someone who has tested positive for COVID-19. We’ve been running fast on that, so that’s been an exciting extra second job.

Ah, yes, Android’s COVID exposure notifications rolling out to a smartphone near you. The APIs for this have shipped in Google Play Services and rolled out to basically all of the two billion Google Play Android devices out there. Full OS updates might be dependent on your device manufacturer, but Play Services updates hit every phone that has the Play Store installed, so basically every Android phone outside of China.



Enlarge / Google is going to start generating COVID apps itself so states and other entities can get up-and-running easier.

The current problem with COVID tracking is that individual health organizations need to make apps that plug into this API, and in the US, that's usually your state government. States don't really have competent app developers on call for something like this (Is your local DMV website as much of a disaster as mine?). So far, only six states currently have COVID apps. It sounds like Google's next step to try and fix this is to make apps itself, so states only need to supply a configuration file to get up and running.

Project Mainline updates

Project Mainline, aka "Google Play System Updates" is one of the biggest changes to come to Android in some time. The feature debuted in Android 10, and it's a major step in the modularization of Android. Mainline added a new "APEX" file

type, designed to package core system components for easy updatability through the Play Store. Previously, the Play Store only shipped APK files, which, since they were built with third-party apps in mind, came with all sorts of security and functionality limitations that wouldn't work for core system components. APEX files can only come from Google or your OEM, so these offer a lot more power and start-up earlier in the boot process. APEX files can do things like update the app framework, which you could never do with an app.

Mainline was also about getting OEMs on-board with Google taking over more of Android's base code—code that previously was available to OEMs to customize. Google had to sit down with all the OEMs (I imagine these meetings look like the Galactic Senate) and ask, "Do you really need to customize the way the DNS resolver works?" When all the answers come back "no," that becomes a Mainline module and everyone agrees to ship the same piece of code. When there are customization concerns, Google and OEMs are working together to upstream code into a module that everyone can use.

That was Android 10. For Android 11, the last news we got about Mainline was Google's first developer preview blog post from February, which said there were "12 new modules" in Mainline. It didn't provide much more detail than that.

Ars: Your blog post said there were "12 new apex modules" in Android 11, but what are they exactly?

Burke: Yeah, there's a bunch. I have a list here: so statsd, which is our daemon for collecting stats, and that makes sense because you want to have uniform telemetry. Wi-Fi tethering is a new module. NNAPI—the neural networks API—again, that's another space that's changing rapidly as we learn more techniques in machine learning. ADBD. Cell broadcast. There's some Wi-Fi modules. SDK extension stuff. Oh, and a media provider as well, which underpins scoped storage, so we wanted that to be updatable.

I think there's a total of 21 modules now and I think probably more important than what the actual modules are is the work that's gone into the infrastructure to make them possible. We have very advanced release management. We've got short-term, long-term telemetry. We've got A-B capability. We've got a file system snapshot in the rollback. And the other part, of course, is just a cultural change for the developers to learn how to write in a module that's being updated all the time. I'm pretty excited about the foundation that we've laid more than what the specific modules are because there's more to come.

Ars: Speaking of "more to come," I wanted to ask about that "SDK extension module," which sounds pretty important. Is this as interesting as I'm imagining? You want to deliver new API levels via the Play Store?

OK, time out while I explain this question: Android versions are identified to you and me by their version numbers, but internally Android identifies itself to apps with a number interchangeably called the "SDK level" or "API level." So all the new features, permissions, and security restrictions in Android 11 are available to apps in "API Level 30." In the past, API Levels have always gone up +1 with each Android release (even for the smaller 0.1 releases, which is why we're at level 30).

The speculation with an SDK module is that Google would be able to ship entire new SDK levels to developers, including new features, without having to push out an entire OS update. This would be absolutely incredible for Android, since full OS updates have such poor distribution and small user bases that developers are reluctant to support new APIs when no one can run them. API levels over Google Play would be just like a Play Services rollout, where a new feature can hit two billion devices virtually overnight. This also sounds very hard to believe, because a new developer API can hit any part of the OS. How could you possibly update that via a single module?

Burke: I think the whole idea of updatable OS modules is a pretty profound shift, so it's all pretty interesting. But yeah, we have the ability in Android 11—Android R, as we call it—to create new system APIs and deploy them in mainline modules. That's in R. In S *[Android S would be version 12],* we're going to plan to be able to actually deliver new public APIs in Mainline modules, so we're really just extending the breadth of what's a module and what's updatable.

Ars: That's going to have to be more limited than a full OS update, right? How well can that work compared to an OTA? It sounds amazing but also pretty hard.

Burke: Yeah, I think it's still early. You're right. The challenge with updatable modules is that the module updates but you can't assume that everything around it updates. So, you have to be careful and have stable internal APIs or boundaries between those interfaces.

So yeah, we're still working. I think what you really want is for the API to be connected only to another updateable module, otherwise it doesn't quite make sense. We're building out the capability and then we'll see what we'll use it for.

https://arstechnica.com/?p=1703393