



Fred Ehrsam [Follow](#)

Previously co-founder @Coinbase, trader @GoldmanSachs, computer science @DukeU.  
Jun 27 · 5 min read

# Scaling Ethereum to Billions of Users

Tokens are selling at valuations which imply they'll have millions of users. But can the blockchain support it? If not, how far away are we?



## Requirements to building scaled decentralized applications (dApps)

There are two requirements to building scaled dApps:

- 1. Having all of the necessary parts of the dApp developer stack. I'd say we are 70% to a shaky first version of each.

	Web 2.0	Web 3.0 (dApps)	Status
Computation	Amazon EC2	Ethereum, Truebit	In progress
File storage	Amazon S3	IPFS/Filecoin, Storj	In progress
External data	3rd party APIs	Oracles (Augur)	In progress
Monetization	Ads, selling goods	Token model	Ready
Payments	Credit Cards, Paypal	Ethereum, Bitcoin, state channels, 0x	Ready

- 2. Scalability of these components. I think this will be the main limiting factor and what will be discussed in this post.

## Facebook as a measuring stick

Let's do the math on running something like Facebook. Facebook handles about 175k requests per second (900k users on the site in any given minute, assume an action is taken every 5 seconds). And this probably doesn't include API requests, which are a better analogue and probably 3-4x higher.

## Where is Ethereum now?

First, a note: I am not an Ethereum maximalist, but I think Ethereum is the furthest along and best suited for thinking about scalability right now.

At the moment, Ethereum can handle about 13 transactions per second, which cuts in half to about **7 transactions per second** for tokens (4.7m gas limit, 21k avg gas price for standard txn = ~220 standard txns every block, current avg block time 17s = 13 txns/sec, gas requirement roughly doubles for token transactions). And this doesn't include more expensive smart contract execution.

By this estimate **we're roughly 250x off being able to run a 10m user app and 25,000x off being able to run Facebook on chain**. And since these systems are open rather than proprietary, we'll see applications bigger than Facebook. This estimate isn't perfect because the dApp stack functions differently in places than the current Web 2.0 stack. But I believe it's in the right ballpark. This is why I believe **scalability will be the primary bottleneck for the industry for the foreseeable future**.

## Scaling to billions of users

So how do we get closer to supporting billions of users? Here's a summary of the scaling efforts I'm aware of:

This high level view makes a few things clear:

- **These are hard computer science and game theory problems. Most of them have never been solved before. Estimating completion can be hard and can be far off.**
- **Large apps can't run solely on chain and likely never will. They need off-chain scaling solutions.**
- While it's tricky to make accurate estimates when combining scaling improvements, **it's conceivable we could see a 100x improvement by the end of 2018, which would allow a 1–10m user app.**
- Scaling is multidimensional. Different methods address different things. So it's hard to measure with a single and global “transactions per second” metric.
- **These projects have very few people working on them**—in most cases less than 5 people. The people currently working on them are exceptionally talented, so that is not at all a knock on their efforts. If you are a developer getting into the space and want to work on hard problems with large impact, I'd suggest getting in touch with one of these projects. Talk to the Ethereum core devs (for on-chain solutions) by joining the Ethereum Core [Gitter chat](#) or contact [Vitalik](#) or [Vlad](#), get into payment channel networks by contacting [Jeff Coleman](#) or the [Raiden Network](#), or contact Truebit by responding to [their active hiring thread on reddit](#).

**How can Ethereum systematically scale better?**

Finally, it is worth thinking about the how this scaling work can be best supported in the future. Capital was the Ethereum Foundation's biggest constraint in January, and it has prudently managed its finances. However, given recent ETH price changes, the Foundation has over \$150m (750k ETH, \$30m fiat) and expenses are ~\$250k a month. The Foundation can now hire more developers at rates competitive to companies like Google, allowing forward thinking developers to take the plunge. Working in the industry is developing it's own risk spectrum. Working on Ethereum could be similar to working at a Google: lower risk with broad impact right away. Working on a token is similar to working on a startup: higher risk and lower initial impact but higher upside potential. How core protocol work is best funded beyond the initial Ethereum Foundation endowment is an open question, but likely further out.

The community may also want to scale beyond central coordination by the Ethereum Foundation over time. As the chart above shows, this is already happening to a small extent. A simple idea would be feature bounties which the community can request and approve when completed. The community could easily crowdfund \$1,000,000+ and let many teams compete like in the DARPA Grand Challenge that kicked off the self driving car push. It's also possible the Ethereum Foundation moves closer to a decentralized governance model (a DAO) over time, which would have the same effect.

## Conclusion

Everything will be tokenized and connected by a blockchain one day. Scalability is the crux of that journey at the moment. Ethereum is orders of magnitude off from being able to support applications with millions of users at the moment. Yet in true decentralized fashion there are a diverse set of efforts attempting to solve that problem. The biggest bottleneck to solving scalability is the number of people working on the problem. If current efforts are well executed, Ethereum could be ready for a 1–10m user app by the end of 2018.

*Thanks to Joey Krug, Vlad Zamfir, Joseph Chow, Dan Finlay, Simon de la Rouviere, Vitalik Buterin, Chris Dixon, Dan Romero, Ryan Zurrer, Panashe Mahachi, and Matt Huang for ideas and clarifications in this post.*



