We’ll be talking about a blockchain system called EOS.

This disclaimer basically says that we are NOT part of block.one, the company creating EOS code. We’re a separate entity. However, this event has been coordinated with them, so hopefully we'll be talking sense.

And just to make sure everybody is aware of this: EOS has not been launched yet, at this moment it’s not a live a system.

OK, but before we introduce EOS, we’re going to briefly describe the context - the main problems currently facing the crypto-space.

**Scalability**

If we want to host real-life businesses on the blockchain, we need high performance. And what is high performance?

* You need about 20k transactions per second just to do Visa & MasterCard, not to mention all the other payment systems out there.
* Facebook does 50k likes per second, not including all the posts and voting and other actions.
* And for currency trading, it's sometimes a hundred thousand transactions per second, just for one currency pair.

So, imagine putting all of these on a single platform, so they can inter-operate. Well, that would require millions of transactions per second. This is a massive scale.

>And what can the blockchain currently offer? Bitcoin does 4 transactions per second. Ethereum – something between 15 and 30.

>And those two systems are extremely expensive to run. Bitcoin and Ethereum shareholders are spending something like 6 billion USD a year just to be able to process less than 35 transactions per second. In case of Bitcoin, this gives us the cost of 50 USD per one transaction. This is really mad.

>You might say it's just a matter of time before scaling solutions are introduced. But actually, there are only two ways to go:

* You can go off-chain, which is basically the idea behind state channels. If it works (which I doubt) it will only solve problems for basic payments. We need more than that.
* The other option is creating sub-domains within one blockchain or a hierarchy of sub-blockchains. This helps a bit, if you want to run a lot of small apps within the same ecosystem. But if you want to compete with big businesses like PayPal, Uber or eBay, or if you're into currency trading - this solves nothing. For those apps you need enormous processing power on a single blockchain and oftentimes on a single CPU thread.

#### Transaction fees

Right now transaction fees are absurdly high. On average you pay 4 USD for a Bitcoin transfer and 30 cents for moving funds on Ethereum, and this will only get worse once Ethereum hits its capacity limits.

No business can survive this.

Even if somehow the fees do get lower, they will still be unpredictable. If you operate on thin margins, any unexpected rise in transaction fees can kill your business overnight.

#### High difficulty of app development

Where are we today? We have the decentralized computer and a bunch of people trying to build applications on top of it. But actually nothing is coming to market. All we have are just prototypes. Why is that?

It’s mostly because developers don't get to spend their time on building business logic and user interfaces. Instead, they are stuck on figuring out the low-level stuff, things like data storage and inter-app communication.

And because everything is being implemented in the app layer, it's being done within relatively inefficient scripting environment. Which makes it really hard.

Furthermore, even if some generic solutions eventually emerge as a result of this effort, they will be expensive to use, as every line of their code will consume gas.

#### Smart-contracts running amok

One way to tackle this problem is trying to prevent it from happening in the first place. However, software development is an imperfect art, so no matter how much effort you put into checking the code, there will always be a black swan event waiting to happen.

>As we can see from this screen-shot, smart-contract platforms usually do NOT consider this issue to be part of their business model. So, when things go wrong, you, as a business, are on your own.

#### Private key security

People get hacked. And people lose their passwords. There will be more and more unsophisticated users entering the space and they cannot be expected to perfectly protect their private keys.

And if you're hacked on a blockchain app similar to eBay, you lose not only your money, but also you lose your reputation.

#### Inter-blockchain communication

Right now blockchains live in their own silos. Even apps on similar blockchains cannot talk to each other: a smart-contract on Ethereum is unable to communicate with a smart-contract living on Ethereum Classic, even though technically they are identical platforms.

And now let’s take a moment to consider what decentralized applications need, if they want to be successful:

* Your app needs to be scalable and cheap to run. That's pretty obvious. You need to be able to scale in order to have enough users to recover your initial costs.
* You need to have access to freemium models or different monetization strategies. If you want to charge your users that’s fine, but it should be the application choice, not the platform it's running on.
* Your app needs to be available via mobile phones & web interfaces, so that users don't have to download & install anything.
* If you build a financial app, your users might expect confidentiality.
* Your users need account names, no public keys, no fancy cryptographic stuff. And they need a procedure for account recovery, in case they are hacked or just lose their password.
* Your app needs to be upgradeable. You need to be able to improve it to keep your business afloat.
* And finally, your app needs bug recovery, something that's often overlooked in this space.

OK, it’s time to introduce EOS, and find out what it can do regarding all those issues.

What is EOS? The simplest way to put it would be this: EOS is a general-purpose smart-contract platform, similar to Ethereum.

That's a fair definition, but in my view, it doesn't do justice to EOS. I prefer this one: EOS is the blockchain for building commercial scale decentralized applications that are indistinguishable from centralized alternatives.

In other words: EOS makes decentralized apps look and behave like conventional apps.

How does EOS work?

Let’s start with Ethereum. This is what Ethereum actually looks like. We have a bare-bone computer (in this case the EVM) and we have some apps trying to run on top of it.

They have a tough job to do because apart from some basic stuff, like the ability to send funds & maybe store some data, they need to take care of pretty much everything else. It's like building a mobile app on a smart-phone without having Android or iOS at your disposal. It would be pretty hard, I guess.

So what are we missing here?

Right, we need an operating system.

And this is what EOS brings to the table. EOS is an operating system for running decentralized applications.

OK, so now let’s go through the main features of EOS.

There are lots of them, so for convenience I’ve divided them into 6 categories.

**#1 Processing power**

We will start with a very bold claim: on day one EOS will be capable of at least 50 thousand transactions per second. Maybe even more.

>Let’s take a look how this fits into our landscape.

>How is EOS going to achieve that? It's gonna do it in two dimensions:

* Firstly, via enormous speed of sequential processing. EOS uses a solution borrowed from a currency exchange engine called LMAX, which is capable of millions of transactions per second on a single thread.
* On top of that we'll have parallel processing, made possible thanks to clear separation between the things which can be done in parallel and things which cannot. This is an important distinction because not everything can be subjected to parallel processing, e.g. managing an order book of a decentralized exchange always needs to be run on a single thread. Whereas things like validation or authentication can be done in parallel.
* What helps a lot to speed things up, is the fact that in EOS nodes establish consensus over events, not over state, as is the standard in all other blockchains. In EOS state is derived from events. At first this might look like an unnecessary complication, but it actually simplifies a lot of things inside EOS and it allows for powerful optimizations.
* Another helping factor is the fact that in EOS there is no such thing as gas - we will talk about it in a moment. As a result, a lot of complexity gets eliminated, as we don't need to count all the tiny operations performed inside the VM.

**#2 Built-in governance**

* In the heart of EOS there is a consensus mechanism called DPOS, which stands for Delegated Proof of Stake. We don't have enough time to go into the details, so I'll just do a quick overview.

Basically, in DPOS there is a fixed number of block producers (around 20) and all of them are elected by the shareholders, each voting according to their stake.

The whole thing is powered by reputation: it’s hard to get elected, but very easy to lose the job. As a result, DPOS allows for very efficient decision making, while ultimate power always rests with the shareholders.

* In EOS the assumption is that no matter what, bugs will be happening, and we need to handle them.

To this end, block producers will have the ability to freeze & fix broken apps. Of course, only in case an app is unable to recover from a bug on its own by upgrading itself. Had the DAO incident happened on EOS, it would have been easily contained within a few hours.

* What else have we got? There'll be a built-in governance mechanism, including a legally binding constitution, an arbitration system for resolving disputes, and an entire system of shareholders voting to decide on things like the level of inflation and the way the income from inflation is being distributed.

Generally, EOS itself acts like a fully autonomous self-governed decentralized app whose business is hosting other decentralized apps.

**#3 Infrastructure for apps**

* In EOS developers only need to write code for what's unique for their application. All common features, including low-level stuff, are provided by the blockchain.

Those features include: account permissions, account recovery, scheduling, authentication, inter-app communication, biometric 2nd factor validation.

Account recovery might be especially interesting here. It works like this: if you get hacked, and lose control of your account, you can get it back, provided you know the previous password, before it was changed by the hacker. And it gets even better: if no hacker was involved but you just lost your password, you can still regain access to your account using EOS social network feature.

* The system has a built-in storage solution based on IPFS and maintained by the block producers, as part of their job.

**#4 No transaction fees**

* This is my favorite. Your EOS tokens are never consumed, as there is no such thing as transaction fee or gas payment for running a smart-contract.
* As a result, if you're an EOS token holder, you can use the system for free. The deal is very simple: if you own 1% of the tokens, you own 1% of the network, including all its resources.
* However, if you still prefer the Ethereum pay-as-you-go model, you can access EOS resources by renting EOS tokens, instead of buying them. The point is, you have a choice. As a result, you can monetize your app any way you want.

**#5 Publish source code, not machine code**

* This one looks quite innocuous but actually it's a big deal. Firstly, it means is that the source code of your app can be recompiled in the future. This way your smart-contract becomes upgradable, so you are no longer stuck on the very first version of your app.
* And secondly, it means that EOS will be able to have multiple VMs and your contract will be able to switch from one to the other.

**#6 Asynchronous communication**

* EOS works like a giant email server, as everything is based on sending and receiving messages. Which means that the entire communication in EOS is asynchronous, the exact opposite of Ethereum’s architecture.
* And it just happens that asynchronous communication is the only way to arrange communication between blockchains. As a result, EOS apps can communicate with each other NOT only within the same blockchain, but also across blockchains. So, if we had multiple EOS blockchains, both public and private, everything could be fully interconnected

Here are EOS main features listed together.

And now I'll tell you a little bit about EOS origin.

There are two ways of building a smart-contract system:

* Either you start by creating the damn thing and then try to use it to run concrete apps on top of it. We might call it a top-down approach.
* Or you start with building a couple of concrete, non-trivial apps, make some mistakes when doing it, learn from those mistakes, and as a next step try to figure out what's common between your apps, and only then start building the abstraction layer. So it’s a bottom-up approach.

You can make your own judgement which one is more likely to be more effective.

Unfortunately, the prevailing approach in the crypto-space is the first one, top-down. Teams of very talented people are creating abstract smart-contract systems, and then they are confronted with tough reality when they try to build something useful on top of those abstract systems.

EOS goes the other way, bottom-up. The team behind EOS has already built some of the most successful blockchain apps in the crypto-space, and now they have set out to leverage their experience by building EOS, a generic platform for hosting such apps.

And when I say most successful apps, I don't mean market capitalization, I mean how popular those apps are among actual users.

This is what it looks like. BitShares and Steem were created by the same developers who are now behind EOS.

And those two apps are very different: BitShares is a decentralized exchange, whereas Steem is a social media platform.

As you can see, Steem is the most used blockchain app in the entire space. Its users perform something close to a million transactions per day, which is still less than 1% of its total capacity.

This is what their UIs look like. Those are very complex apps. They still have their deficiencies, but they work smoothly enough to trick you into thinking that they are just normal web apps.

They have a couple of amazing features, e.g. account recovery was first introduced in Steem. Both use the LMAX-inspired sequential processing engine, we’ve just described. So a lot of EOS features originate from those systems.

And one more interesting thing: while those apps are quite popular among users, in reality none of them could run on any existing smart-contract platform, including Ethereum.

Why is that?

Firstly, because those two little apps alone would eat up all of Ethereum processing power.

Secondly, Steem business model requires that all user actions are free of charge (as otherwise nobody would upvote anything). Ethereum is unable to offer that and most probably never will.

Actually, there is only one smart-contract platform able to host an app similar to Steem and I guess you now know which one it is.

For reasons unknown to me, there is an ongoing controversy regarding DPOS, the consensus system, which will be applied in EOS. Some people believe it’s not decentralized enough.

Which is quite strange, as DPOS is arguably the most decentralized system in existence, provided we apply a rational measure of decentralization: how many unique entities are involved in producing 80% of the blocks.

This is the data for BitShares & Steem, both of which are using DPOS.

Furthermore, DPOS is extremely resilient against any attempt to shut it down. It can adapt to very tough conditions and still survive, and then fully recover. This is mainly because it's very easy to replace non-performing block producers.

By the way, DPOS is relatively cheap to operate – it doesn’t cost billions of dollars per year. As a result, the inflationary funds can be utilized for other goals.

EOS strong points:

* EOS has a very efficient team with a proven track-record. Steem was invented, created and launched in just four months. And this is probably the most complex blockchain app currently in production.
* EOS is using Web Assembly for running smart-contracts. Web Assembly is a big deal - it's an emerging industry standard for web applications supported by Google, Microsoft, and Apple. You should look it up, but basically it's a universal compile target for several programming languages, including C++, C#, Java, JavaScript and possibly even Solidity.
* Regarding EOS financial supporters, we have some big names here: Brock Pierce and Bo Shen. Interestingly, both of them were also early investors of Ethereum, and still are.
* And last but not least, we have a declaration from EOS founders to spend about 1 billion USD on creating an ecosystem around EOS. Apparently, this is where most funds from the currently running ICO will go - not on EOS code development (as this is already fully funded) but on those of you who will try to do something useful with EOS.

Unsurprisingly, EOS has also got its weak points.

* The ecosystem around EOS has not really emerged yet. It might be hard for EOS to catch up with the high number of developers currently working on Ethereum-based projects.
* Track-record of poor documentation. Unfortunately, this is true. Those guys behind EOS are brilliant, but their projects have always lacked good documentation. Hopefully it will change this time around.
* Using C++ is right now the only way to create smart-contracts on EOS. This will surely change, as Web Assembly gets upgraded, but unfortunately at this early stage this might be a big turn-off for developers who are unfamiliar with C++.
* And it's not live yet, but it will be in June next year.

This is the roadmap. Currently we are at the MVP stage.

You can already play with the system on your local machine and start building smart-contracts with it, provided you have some basic knowledge of C++.

In a couple of days there'll be a subsequent release, called Dawn 2.0, and there’ll be a public test-net up & running.

And now let's revisit our list of requirements for decentralized apps. As you can see EOS is doing a pretty good job here.

At this stage, I’m not aware of any special features in EOS regarding zero-knowledge cryptography or untraceable transactions.

And we can also revisit our crypto-space problems. Again, as you can see, EOS features nicely match just about all of them.

I hope all of this is self-explanatory.

To wrap it up: I think that what sets EOS aside, is not just the technology - it's actually the way it's going to operate.

* EOS builds on solutions which have already been battle-tested in real-life blockchain apps. So it looks like a revolution, but actually it’s an evolution.
* Whereas Ethereum is an excellent playing field for smart-contract prototypes, EOS takes the notion of smart-contracts to the next level and makes the whole thing extremely business oriented.
* Can other systems copy EOS solutions? Not really. EOS is quite different from anything else on a very fundamental level. To follow EOS path, other platforms would have to backtrack a lot regarding the fundamental elements of their technology. So, it's very unlikely.

And finally, this graph shows the conclusion we have reached when observing how blockchain technology evolves.

Back in 2008 we started with a simple payment system, which then evolved into a smart-contract system, which now is about to evolve into a fully blown operating system, not just for smart-contracts, but for entire blockchain-based apps.

Will EOS replace Ethereum? I don't think so, at least in the predictable future. But what will probably happen is EOS playing the same role for Ethereum, as Ethereum is playing for Bitcoin.

And now very briefly about our company:

* Our focus is on blockchain-based fundraising and digital asset management solutions.
* We are in the process of setting up a software house dedicated to building dApps, both on Ethereum and EOS.
* And we are aiming to be elected as EOS witness, which is just another name for block producer. We have some experience with that, as one of our advisors is already a witness for Steem.