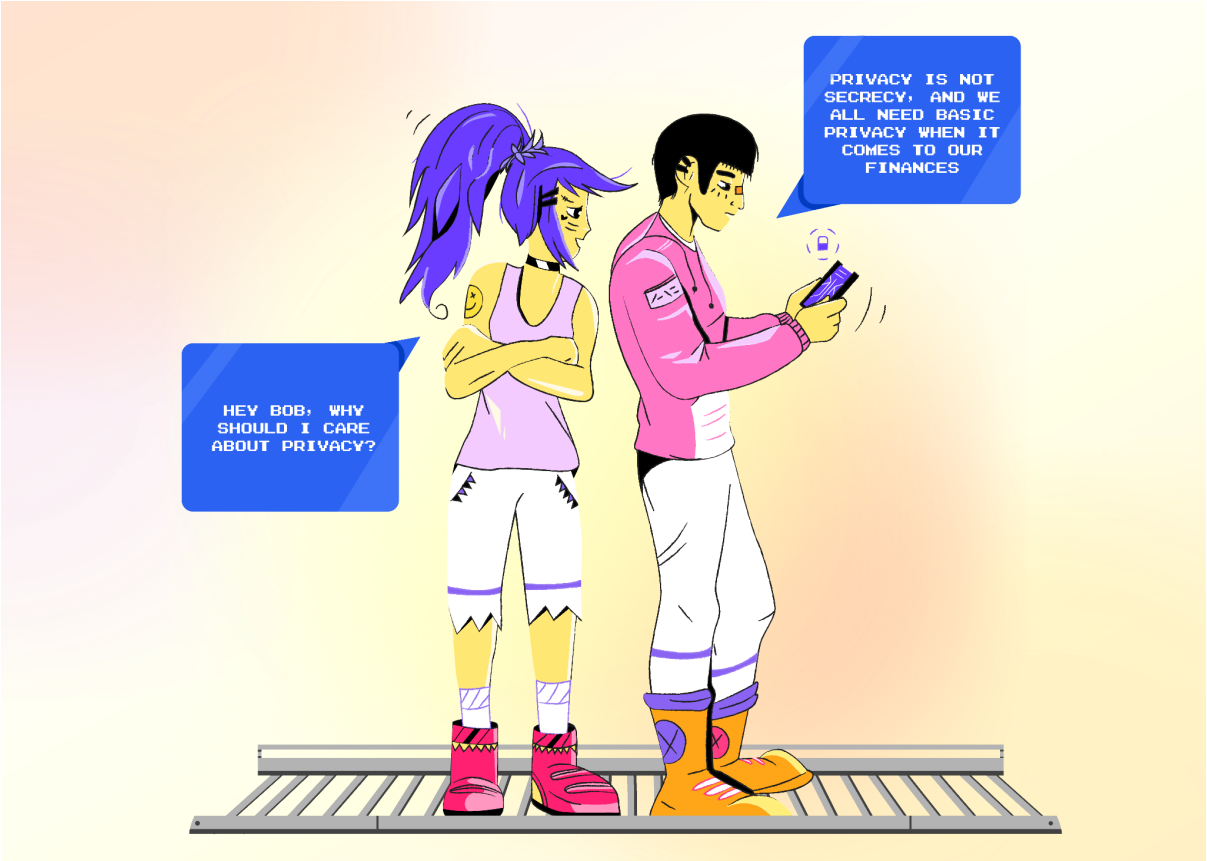
ZkBob: your friend and guide to transact in private



Alice: Hey Bob, what's all this about? Why should I care about privacy at all? I have been using crypto forever, since 2020. Everyone says that it's anonymous. Is it really? If it's not, maybe it's a good thing? Why keep things secret if I have nothing to hide?

ZkBob: Privacy is not secrecy. We all need basic privacy to feel safe and comfortable. And when it comes to money, we should be able to choose privacy and disclose details only when we want to, or when we're obligated to.

In the Web2.0 world we've gotten used to trusting centralized authorities to securely store and use our data in exchange for free services. But somehow Web3 has removed trusted intermediaries and eliminated our ability to transact privately. Although Bitcoin started with a notion of anonymity via burner addresses, most modern decentralized applications do not support stand-alone confidentiality.

There are two main reasons why:

1. Transacting in a private manner is commonly associated with hiding illicit activities.
2. Most current Web3 activities revolve around trading applications.

Alice: Yeah, it does seem like trading and DeFi are still the most popular ways people are using blockchain, but I have seen more businesses interested in moving operations to a web3 environment.

ZkBob: There is a lot of interest there, but right now regular business use cases like automating payroll or paying taxes are not very workable. Competitors and other employees can see salaries, payments, inflows and outflows, hackers can monitor employee wallets for phishing… it's not good for a lot of typical business operations.

If we're really trying to push Web3 adoption into the mainstream we have to somehow solve the dilemma of maintaining privacy while remaining compliant.

Alice: Ok, basic privacy rights for individuals and for business makes total sense. But how can I get more privacy when everything is completely transparent on the blockchain?

ZkBob: Glad you asked…it's called zkBob! The zkBob application uses a special stable token called BOB and cryptographic constructions known as zkSNARKs to help you achieve privacy.

You can get BOB on-chain, then use the zkBob app to convert your BOB stable tokens into a shielded version. With the shielded tokens, only you know the balance and only your counterparties (the people you transact with) can see the corresponding operations. The only data posted on-chain are cryptographic commitments that ensure balance integrity, but don't reveal any details about the transaction itself like the sender, amount, or receiver.

At the same time, different tiers for transaction limits, real time monitoring of incoming funds, and other safeguards mean the zkBob private pool is uncontaminated by illicit funds - it won't include stolen funds from hackers or other shady business.

Alice: Wow, that sounds amazing, especially for businesses setting up payroll systems or even DAOs and web3 contractors! But how does it actually work?

ZkBob: To understand how zkBob works, let's first look at deposits and shielded funds.

We'll start with a deposit, which is actually a conversion between an ERC20 stable token BOB and its new shielded form within the application.

When you make a deposit:

- Your ERC-20 BOB tokens are locked inside the Pool contract
- Once locked, you can manage these shielded funds within the zkBob application.

Locking the tokens is somewhat easy, but managing shielded funds is more complicated. To make an analogy, imagine that each deposit creates a new safe box. The box stores the funds and can only be opened with the key held by the depositor. Furthermore, the box can only be opened once.

The key to open the safe box is called a nullifier, and when you want to make a transfer or withdrawal you use the nullifier key to open the box. When the box is opened using this nullifier key, an identifier is revealed that proves the key has been used. Following your one-time transfer or withdrawal, this key and safe box cannot be used again.

So getting back to the deposit, here is what happens:

- You generate a key chain via a fresh seed phrase or existing MetaMask identity, whichever you prefer.
- You create and sign a new nullifier with your onchain identity, ensuring that your funds go directly to your own safe box and nowhere else.
- The transaction is processed on-chain, meaning funds from your account are locked in the Pool contract.

Alice: Ok, but how do I know that I really own those shielded funds?

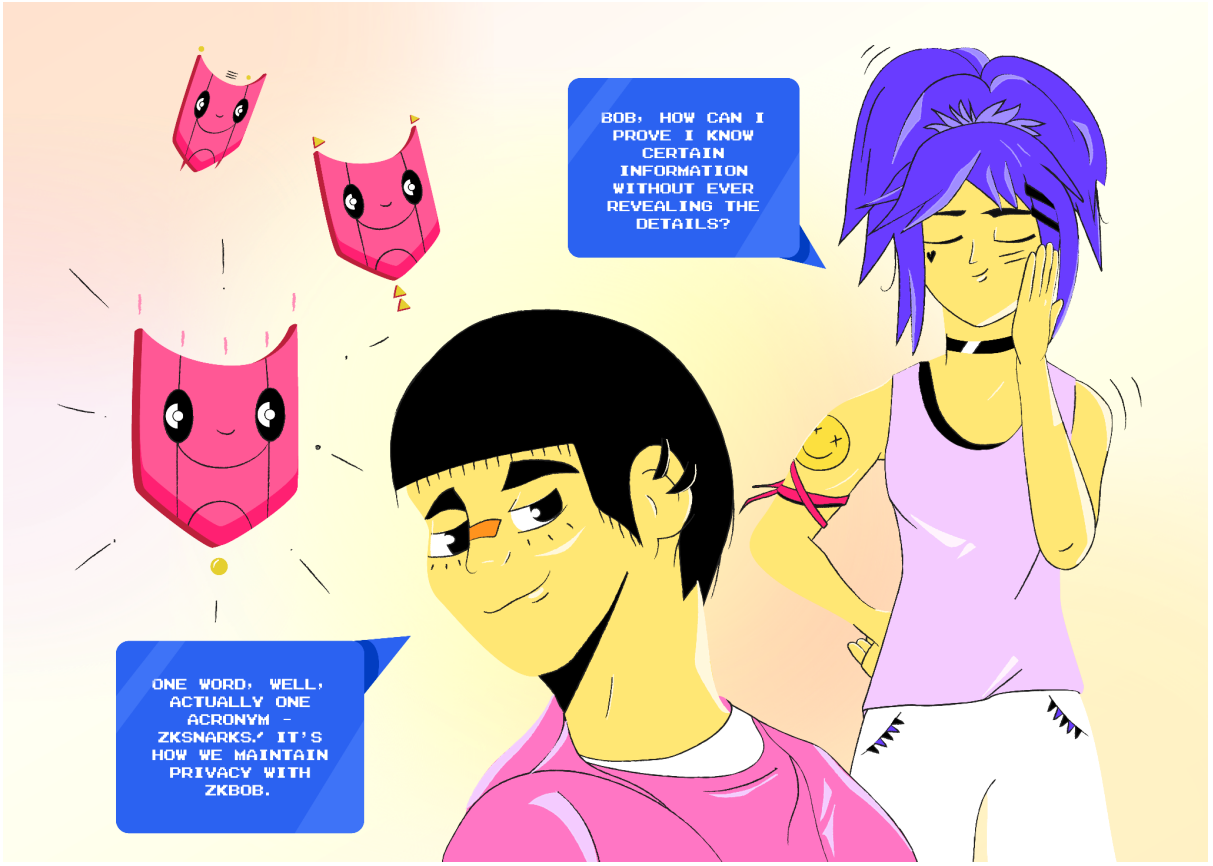
ZkBob: One way you know you own the funds is that you have the ability to do something with them - for example you can transfer or withdraw funds from the app. You know you own them by the fact that you can use them. With zkBob, you actually don't need to specify a box when moving funds. You can transfer funds between safe boxes without ever specifying the box that is used!

Alice: So I can transfer funds to another box (someone else's account) without saying which box it is?

ZkBob: Yes! If the prover (the party initiating the transfer, you in this case!) can prove they know certain information, even without directly disclosing it, the transfer can be processed. To complete a transaction, you must be able to prove that you:

1. Have the key to an existing box.
2. Have never used the key (therefore the same box) before.
3. The amount of tokens in the old box (box you are transferring from) is greater than or equal to the amount you are requesting to transfer (including fees).

Alice: But how can I prove these things without specifying a box where they should be sent?



ZkBob: One word - well one acronym actually - zkSNARKs. A zkSNARK is a "Zero Knowledge Non-Interactive Argument of Knowledge". Now that's a lot of knowledge!

A SNARK proof itself is a pretty intricate thing, and I won't go into details about the Groth16 protocol we use in zkBob. But in a nutshell all of the actions you perform locally, on your own computer, are encoded in a special way to a mathematical function (a polynome technically, but that's not the point). The correctness of that function is checked against a "question" that was agreed upon in advance by the protocol.

You could imagine it as trying to prove to someone that you've read a very large book by answering just a few questions about the main characters. With a zkSNARK, the question itself, as well as the answer, is encrypted. This makes it impossible to cheat. You can prove you know the answer, but the answer itself is never revealed.

Alice: So what do I have to prove to be able to make a transfer?

ZkBob: In this case, if prove that you:

1. Have the key for the box.
2. Know the box "location"
3. Know the contents of the box.

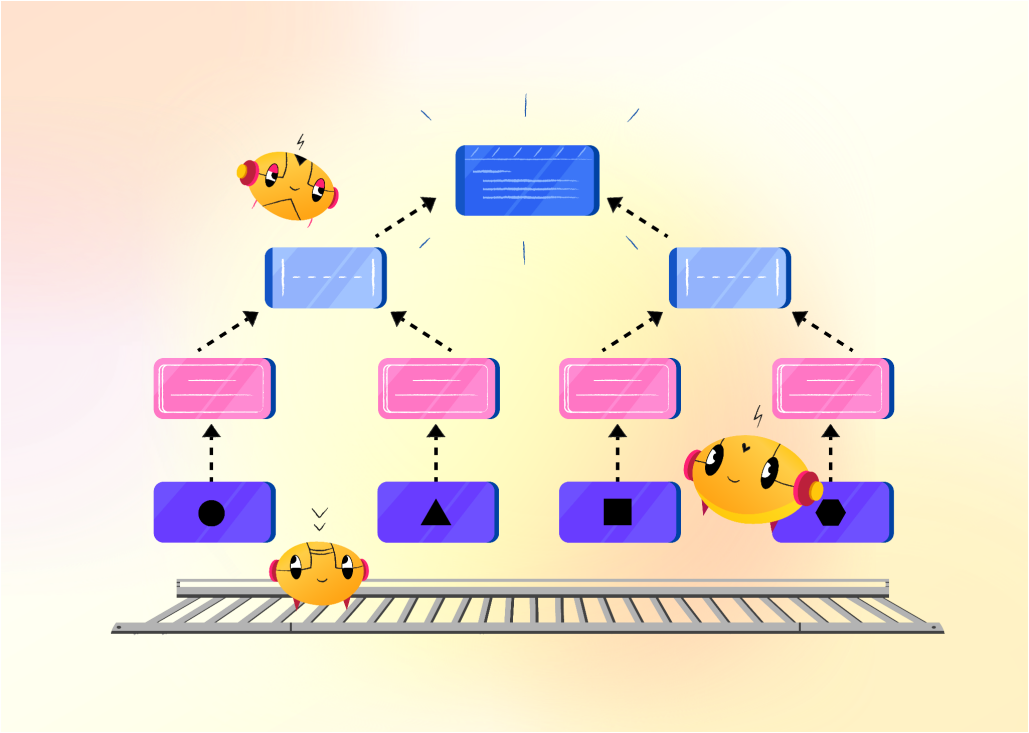
If you can prove these things, the application will be convinced that you own the box and have the right to transfer funds from it, without specifying the exact box.

Alice: And what do you mean by box location?

ZkBob: Here we use something called a Merkle Tree. It's a way to store lots of information in a compact way. With a merkle tree we hash multiple values to a single "root" that contains every value location (index) in the list.

These values are referred to as "leaves". Leaves are then linked in pairs and hashed together. Going up the tree to the next level, the hashes are linked and hashed again. This continues upward until a single value (the root) remains. If you imagine these steps in an up-to-down way it looks like a tree, although it's mostly illustrated upside-down with the root at the top.

So each leaf has a unique path to the root, consisting of the hash of the leaf itself and all of the hashes of the nearest siblings at every level. This is called the Merkle path, and you can use the root to prove the existence of any leaf that corresponds to that root hash.



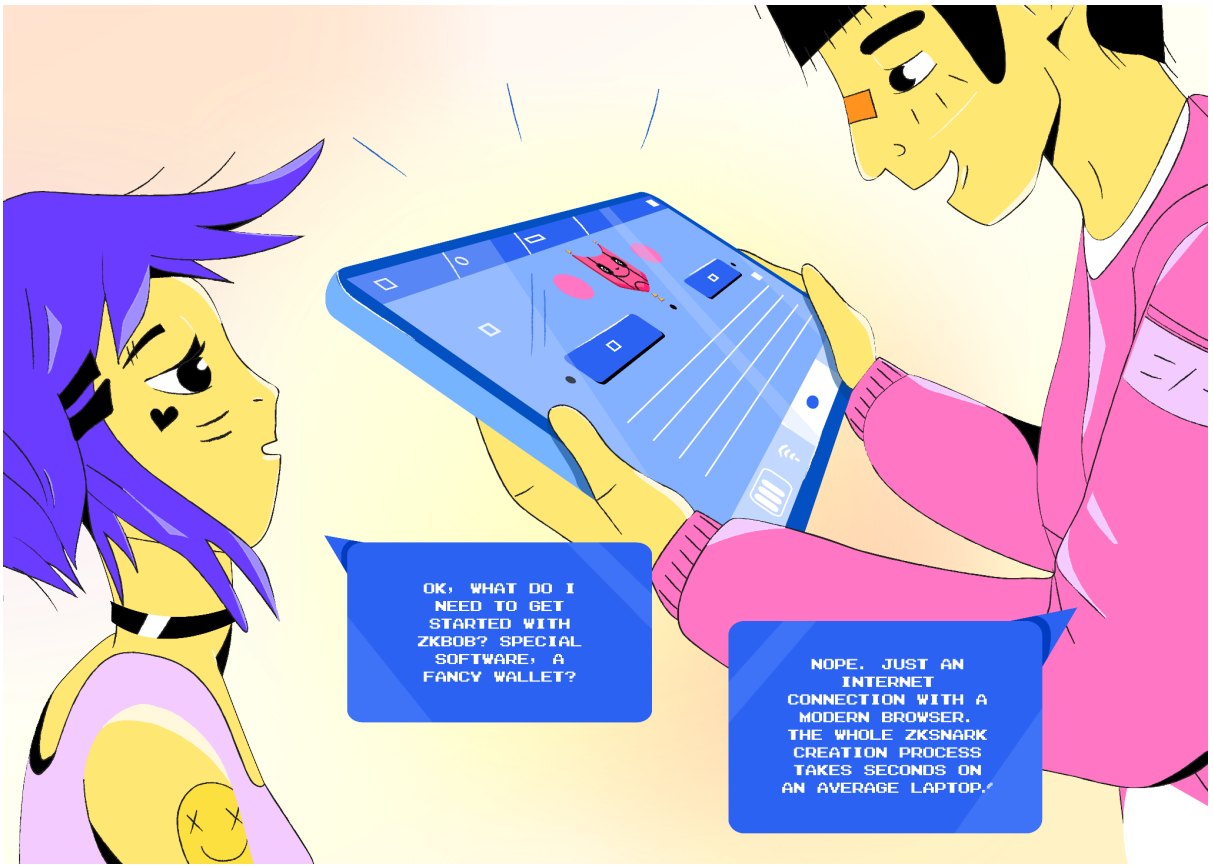
Alice: Ok, so a leaf in the merkle tree is basically the same as the box location, right? But how does that fit into not specifying the box, or to maintaining privacy?

ZkBob: It comes back to zkSNARKs**.** When you initiate a transfer, you calculate a valid signature with a key and a valid Merkle path locally in your browser, add the details for the new safe box, then generate a SNARK proof that proves everything was done correctly. It proves that you have the required information and own the account.

Alice: Going back to that “Question” you mentioned before? If I don’t know it, then someone else does, right? So someone else can take my funds…

ZkBob: Ok, we’re entering some pretty heady territory, I’ll keep it brief here and you can always check out more details in the docs! The “question” is also known as a “challenge”, and the challenge is computed in a multiplayer setup process with “the ceremony” and a “trusted setup”. The ceremony is conducted, and the challenge is created, when the application is first setup. The main idea is that due to the fact that there are multiple participants who blindly contribute to the process, we can trust that the setup process and secret information is safe (as long as there is a single honest participant among the contributors).

Alice: Okay, so let's assume for now that it actually works! But how do I do all of this? Is there special software or hardware? A special wallet?



ZkBob: Thanks to modern browsers you can use a web application which stores your keychain in a securely encrypted way and the application uses it to perform all of the computation needed to make a transaction. The whole zkSNARK creation process takes seconds on an average laptop. There are some limitations however, like you can't really use a hardware wallet, since they lack resources for some heavy math. Mobile browsers can also be a challenge, but I'm working on it!

Alice: And then what? Do I use MetaMask to post a transaction on blockchain? Do I have to pay for the gas?

ZkBob: To deposit you can connect Metamask or another web3 wallet to first deposit BOB into the application. After that, you don't need MetaMask at all to interact with the application. And, you won't pay any gas fees yourself, on deposit, transfer or withdrawals. Paying specific gas fees can compromise your privacy, so all fees are paid in BOB, and each transaction costs exactly $0.10 BOB.

Gas fees are paid by a special service called the relayer. The relayer posts transactions on your behalf and also handles additional work itself like helping prove that the Merkle Tree was calculated correctly.

Alice: So do I depend on the relayer somehow for transactions? Can my funds be compromised by that service?

ZkBob: The relayer helps facilitate transactions, but the most important thing is the relayer has absolutely no control over your funds. The only thing that could happen is a denial of service because of an incident, and that can be easily mitigated with more robust architecture and additional redundancy. Down the road I'd like to make the relayer a decentralized mesh-like service with opt-in participation - in fact it's on the roadmap!

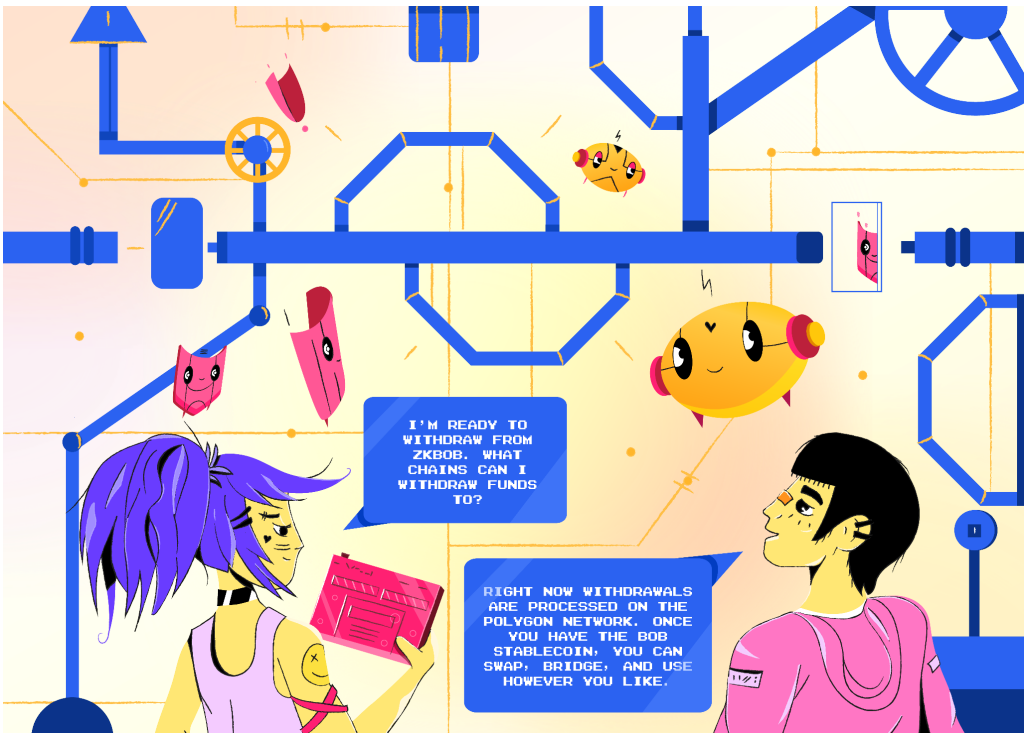
Alice: Does the relayer know who the transfer recipient is?

ZkBob: No. No one except you and the recipient know that information.

Alice: But what if I do multiple operations? Will all of the other people I've transferred with see the connections between them?

ZkBob: No again! Every time you receive some funds you can generate a fresh "receiving key", so that there is no way two parties could find out they are linked. And with a single receiving address no one can deduce from the transaction list or any other publicly available information anything about your other assets. It really is private. What makes this privacy pool work though is a large anonymity set, which is just made up of regular users using zkBob. The more the merrier, so please do invite your friends!

Alice: I will! I love the idea that you can transfer easily while keeping basic information private, all on the blockchain! One more question - once I've finished transferring, how do I withdraw? Can I withdraw my funds to a different chain?



ZkBob: When you withdraw you specify an address on the Polygon network that the withdrawal is sent to and press withdraw. It's that easy! You can withdraw to a brand new 0x address you create without any history to maintain privacy, or it can be any other address you want to withdraw to, including a hardware wallet or multisig.

Right now you can't withdraw to another chain directly. I've started to look at some bridge implementations that would make this possible. Of course once you withdraw BOB, you can trade it for another asset on Polygon and use existing bridges to move it to other chains.

Alice: Cool. I think I'm ready to give it a try! I know my company will be interested in trying it out for payroll purposes, it's the perfect fit. Are there other use cases apart from payroll where zkBob makes sense?

ZkBob: Payroll is the first primary use case because it's a major pain point for many organizations who want to use the blockchain to pay their employees and contractors but haven't found a secure and private way to do so. Beyond that, there are lots of other use cases zkBob can help with.

Businesses might use it not just to pay employees but also to pay vendors, so that their costs and business models are protected. Donations are another good use case — sometimes individuals or groups want to donate anonymously to different causes without exposing their identities or affiliations. And how about plain old transfers between friends for a dinner out, or a gift. There are so many transactions we make every day that can benefit from a bit of privacy and don't need to be broadcast to the entire world!

Alice: This has really made me think more about privacy. The way I choose to spend my money, how much money I have, and how much money I make are private things. Like you said at the beginning - privacy is not secrecy. We all need basic privacy to feel safe and comfortable. I don't need advertisers knowing what products I bought, or my neighbor seeing how much I get paid every month, or any stranger diving into the details of every single transaction I make on the blockchain. Basic financial privacy feels like a basic right.

zkBob: Yes, I believe it is a basic right for all people. Unfortunately, bad actors have misused some of the tools that do exist out there for privacy, making it more difficult for honest people. I want to create an environment where people can transact privately in a safe environment. Transaction tiers and funds monitoring are helping create that environment with zkBob. I also want privacy to be easy for anyone. I'm going to keep on working to make zkBob a simple app to use and understand so we can keep bringing more people and businesses into the web3 world!

Alice: Thanks Bob! I appreciate your time and passion for everyday privacy!

zkBob: For sure Alice, and please do send me a private DM with a single receiving zkAddress, I want to send you some BOB!