



# **SEP-30** and the Importance of Key Management and Recovery

Leigh McCulloch  
Principal Software Engineer  
Stellar Development Foundation



Hi everyone,

My name is Leigh McCulloch and I'm a software engineer at the Stellar Development Foundation.

I'm working on improving the experience that individuals have using Stellar, and that businesses have building on Stellar.

Today I'm talking about the importance of key management and how it defines the user experience of a crypto product.

I'm also going to talk about Stellar Ecosystem Proposal 30 and how it can give wallets an end user experience that is comparable to consumer financial products.



2

When we talk about key management it's critical we are on the same page of what that means, so we're going to start there.

At the core of owning a Stellar account, or funds on many blockchains, is a key.

On the Stellar network if you have an account you have a key for that account.

**TRANSITION:** But that key doesn't look like this cute key, it looks like...this.



**G**DBB4A5CNKPXZOHN  
DKFH5RLY7MHJMBP7  
5FPLR2Y4F63OMDHA  
3XBAKUQ3

3

**TRANSITION:** It looks like...this.

A key is made up of two parts a public key that you share with others and a secret key that you keep private.

This is a Stellar public key.  
It's easy to identify because it starts with a G.

The public key for an account is also the account's address.

If anyone send funds to this address and they will show up in the account's balance.



**S**DZ30KG7TOIKKBGN  
HJFQB5TLP HBRSB3Z  
UROZU30KAXZB2GG2  
A7JIWAKV

4

This is what the secret key looks like. It starts with an S.

On the Stellar network we call the secret key that defines your account address the accounts **master key**.

If you create a new account on the network and don't change its configuration, your master key is what you use to control your account.

If you're using a Stellar wallet today you might be thinking, you've never seen one of the S secret keys before. There are a few reasons that could be the case. We'll come back to that.

It's worth us talking concretely about what it means to control an account.

**TRANSITION:** So let's have a look at how a key can control an account...



Pay to	
GSADIG7TOIJRP3IY47MK...	
Words	Amount
Five	5.00
Memo 	Sign SDZ30KG7TOIKKBGNHJF...

5

A secret key controls your account by signing transactions.

It's similar to how signing a check works.

A check specifies an amount to be paid from the payer to the payee.

And a Stellar transaction specifies an amount to be paid from a source to a destination.

Like a check has a box for signing with a pen.

Stellar accounts have signers, and signers add a signature to a transaction to authorize it.

The default signer is the secret key that defines the account address, the accounts master key.



But a signer is just another key that can authorize transactions for a Stellar account.

If you get a new account, it's got one signer and that signer is the master key, but...

**TRANSITION:** Accounts can also have more signers...in addition to or instead of the

master key.



Pay to	
GSADIG7TOIJRP3IY47MK...	
Words	Amount
Five	5.00
Memo 	Sign <span style="background-color: yellow;">SCUWSRQ7LCGLYHZ6L0K...</span>
	SDZ30KG7TOIKKBGNHJF...

6



**TRANSITION:** Accounts can also have more signers...in addition to or instead of the master key.

Accounts can be configured to let signers authorize transactions individually, or together.

On Stellar the configuration for what combinations of signers are required to authorize transactions is controlled by setting a threshold on the account, and weights for each signer.

**TRANSITION:** For example, if you wanted either of these signers to be able to authorize transactions...



Pay to	
GSADIG7TOIJRP3IY47MK...	
Words	Amount
Five	5.00
Memo 	Sign
	SCUWSRQ7LCGLYHZ6LOK...
	SDZ30KG7TOIKKBGNHJF...

threshold: 1

w: 1  
w: 1

7

**TRANSITION:** For example, if you wanted either of these signers to be able to authorize transactions...

You could set the threshold to 1 and give each signer a weight of 1.



Only one signature with a weight of 1 would be required to sign for the account.

This would be similar to how checks for a joint bank account can be signed by either person.

**TRANSITION:** If you wanted both signers to authorize transactions together...





Pay to	
GSADIG7TOIJRP3IY47MK...	
Words	Amount
Five	5.00
Memo 	Sign
	SCUWSRQ7LCGLYHZ6L0K...
	SDZ30KG7TOIKKBGNHJF...

threshold: 2

w: 1  
w: 1



8

**TRANSITION:** If you wanted both signers to authorize transactions together...

You could set a threshold of 2, requiring two signatures to meet the threshold.

This would be similar to how checks from a business bank account might require signatures by two employees.



Pay to	
GSADIG7TOIJRP3IY47MK...	
Words	Amount
Five	5.00
Memo 	Sign
	SCUWSRQ7LCGLYHZ6L0K...
	SDZ30KG7TOIKKBGNHJF...

9

You'll hear people use the term **multi-sig** to refer to accounts configured with multiple signers.

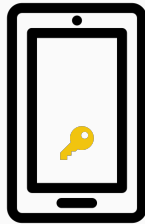
**But**, most accounts on the network use only a single key, the master key, and therefore only have one signer.

Key management is how these secret keys are managed and used, whether it is a single master key, or a more complex configuration.

## PAUSE

Unlike checks, users don't usually manage these keys and transact with them on pen and paper.

**TRANSITION:** And that's where...wallets come in.



10

**TRANSITION:** And that's where...wallets come in.

Wallets are applications or products that help users manage their account.

User's might download the application on their phone or sign up on a website.

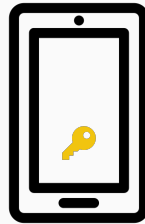
Wallet's play a pivotal role in key management because at the core of a wallet is key management.

**TRANSITION:** A wallet's core job is to...



# Transact

## Store



## Protect

## Prevent Loss

11

**TRANSITION:** A wallet's core job is to...

- Store your key.
- Protect your key.
- Help you use your key to transact.
- Prevent you from losing your **key**.

**TRANSITION:** A wallet is a user interface for key management...and how a wallet implements key management will define the user experience.

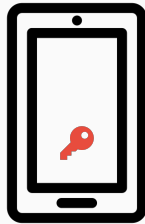


# How a wallet implements key management will define the user experience.

12

**TRANSITION:** A wallet is a user interface for key management...and how a wallet implements key management will define the experience the user has.

**TRANSITION:** Let's look at some ways that wallets define their user experience through their approach to key management.



13

Many wallets use:

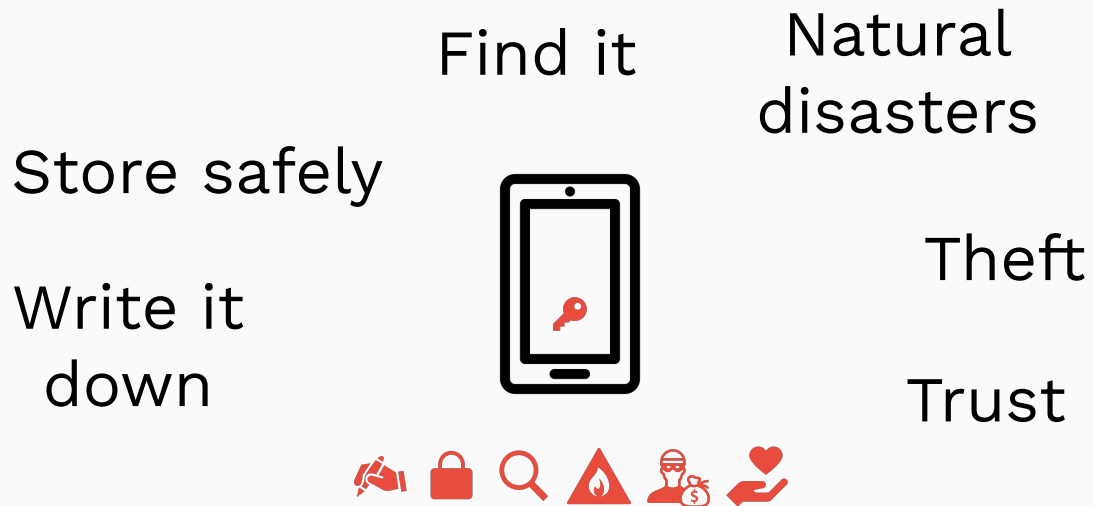
- The master key as the only signer of the account. This means an account has one key and one signer.
- And store your key on a device such as a phone.

It's easy to lose a key on a phone, because phones are easily damaged or stolen.

I've dropped a phone in a lake, and in a toilet... 2 toilets.

So, wallets help you not lose your key using some different approaches which we'll walk through.

**TRANSITION:** For each approach we'll consider...what responsibilities the user is taking on.



14

**TRANSITION:** We're going to look at a few different approaches and for each consider...what responsibilities the user is taking on.

The reason that responsibilities matter is because the responsibilities a user has are as much the user experience of the wallet as the wallet's user interface is.

I'm going to use these icons to represent some of the responsibilities a user might have:

- Writing down the key correctly.
- Storing safely.
- Being able to find it, or not lose it.
- Protecting it from natural disasters.
- Protecting it from theft.
- Trusting someone else to do these things.

**TRANSITION:** The first approach to consider is...



**S**DZ30KG7TOIKKBGN  
HJFQB5TLP HBRSB3Z  
UROZU30KAXZB2GG2  
A7JIWAKV

15

The first approach to consider is the simplest for the product to implement. Ask the user to write down or backup the S secret key, the master key, for their account.

This is simple, direct, and very easy to implement. It's the ultimate in be your own bank. You hold the keys and have 100% control.

**TRANSITION:** With a lot of power comes with high degree of...responsibility.





**SDZ30KG7TOIKKBGN**  
**HJFQB5TLP HBRSB3Z**  
**UROZU30KAXZB2GG2**  
**A7JIWAKV**

17

**TRANSITION:** With a lot of power comes with high degree of...responsibility.

That responsibility is a burden to do everything:

- Write the key exactly correct.
- Store the key safely.
- Not lose it, or be able to find it.
- Protect the key from unforeseen events like your home burning down.
- Protect it from people who might find it or steal it.

The only thing this approach doesn't require of the user is to trust someone else to look after the key.

If a product uses this method with the wrong demographic, user's may drop out during onboarding purely because writing the key down is inconvenient or confusing.

In the worst case users who are not equipped to implement this high degree of responsibility may complete onboarding then experience loss. That experience may create feelings of distrust, fear, or dislike towards the products and brands that didn't set them up to succeed.

We've all heard stories of lost keys or stolen coins and we don't want that.

**TRANSITION:** Another similar approach that addresses some of these shortcomings is the use of...mnemonic phrases, or recovery phrases.



**dignity cross apple cat**  
**scout object fence hen**  
**merge march salon lazy**  
glad canal search crowd  
jealous pledge omit design  
inject march fuel worry

18

**TRANSITION:** Another similar approach that addresses some of these shortcomings is the use of...mnemonic phrases, or recovery phrases.

They are 12 to 24 word phrases that can be used to derive your secret key.

This approach is really popular.

They're easier to write down and they make an application feel friendlier.

But if this is the only method for account recovery, it suffers the same problems.

An account holder is writing down a key on paper and hoping it won't be lost or stolen.

It's 2020. We've been told not to write down passwords, and writing down a secret key or recovery phrases is the same.

For users who are equipped to protect their keys themselves these two approaches might be great. If you are a product developer that is targeting this type of experience that's great. However, we know that products sometimes pick this option because it is simpler to get started, and not because of an alignment with their users.

**TRANSITION:** Products using these approaches could have a slick user experience inside the product, but the user experience of a product...extends out into everyday life. The product impacts a user which includes implementing these responsibilities.



# User experience extends outside of your product.

18

**TRANSITION:** Products using these approaches could have a slick user experience inside the product, but the user experience of a product...extends out into everyday the product impacts a user which includes implementing these responsibilities.

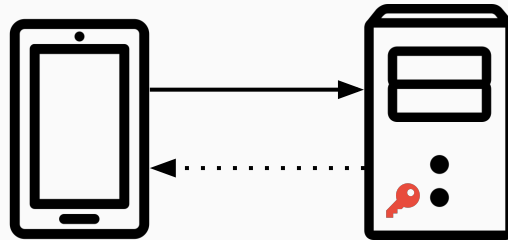
If the user has to protect the key outside an app in the real world, that is part of the user experience.

If the user fails to do that well, that experience of failure is part of the user experience of the product.

## PAUSE

Let's move on to some other approaches that require more effort on the part of the product.

**TRANSITION:** Some products are at the other end of the spectrum when it comes to key management. These products...store and manage your keys for you.



**TRANSITION:** Some products are at the other end of the spectrum when it comes to key management. These products...store and manage your keys for you.

You trust the operator of the product. They perform transactions based on your requests, and you rely on them to keep your assets secure.

The product might hold your assets in a unique Stellar account that only holds your assets, or they might hold your assets in an omnibus account, an account holding all their user's assets together. They might carry insurance, or meet compliance and regulations that give you confidence that they really are holding onto your assets for you.

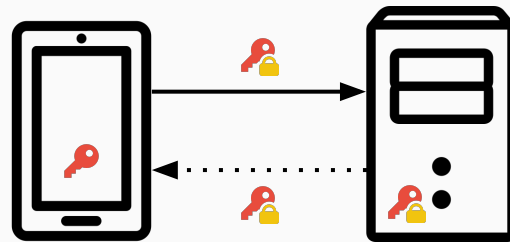
Products like this can create any user experience they desire because the product can absorb key management for the user. They can offer familiar authentication, such as username and password, SMS codes. They can offer account recovery with the option to reset your password, or identity checks like verifying your passport.

## PAUSE

For products where the user manages their own keys, getting this same user experience is desirable, especially if the product is targeting end users who don't have the capability to protect their own keys.

**TRANSITION:** One way that products attempt to do this is to change the thing a user

needs to protect from a key, into a...password.



22

**TRANSITION:** One way that products attempt to do this is to change the thing a user needs to protect from a key, into a...password.

The key is encrypted using a password that the user enters into the wallet but is never shared with the wallet server.

The wallet server is not able to use the key because it cannot decrypt it.

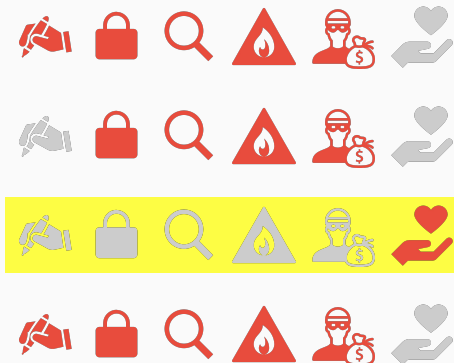
This may feel more familiar to a user because we're already accustomed to remembering passwords for our email and online accounts.

The user experience may appear to be similar to that of a traditional financial services product, but because a secure encryption password is required that can't be shared with the service, this approach suffers some of the same problems as with an S secret key or a recovery phrase.

If the user forgets their password there is no forgot password feature that will recover it for them, and their account will be lost. All the responsibility on remembering it and protecting it is still on the user.

This approach on the surface appears to have a smooth user experience, but if a user chooses a weak password or lose their password, the experience created by the product will be heart breaking.

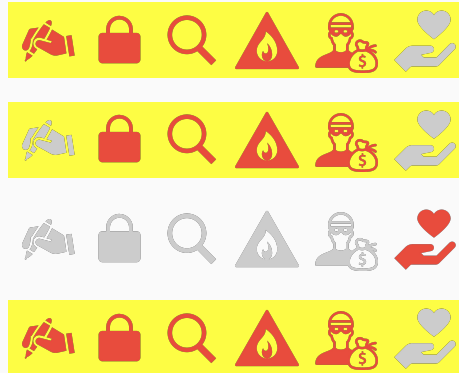
**TRANSITION:** Reflecting on the approaches we've discussed so far...



**TRANSITION:** Reflecting on the approaches we've discussed so far...

The approaches requiring a **low** degree of user responsibility are the solutions that manage the key for the user. The responsibility is converted into trust in the product or third party service.





The approaches requiring many user responsibilities are the solutions where the user manages their key.

The innovations that attempt to make the user experience better, don't actually take responsibilities away from the user.

## PAUSE

At the Stellar Development Foundation we've been exploring how products can change the user experience without shifting trust to a single entity.

We've be exploring how we can leverage the multi-sig features that the Stellar network has built-in.

Multi-sig is the feature we looked at earlier where multiple people can sign transactions in the same way that multiple people can sign checks for a bank account.

**TRANSITION:** That exploration has so far culminated in the...SEP-30 Account Recovery protocol.



# SEP-30 Account Recovery

[stellar.org/protocol/sep-30](https://stellar.org/protocol/sep-30)

23

**TRANSITION:** That exploration has so far culminated in the...SEP-30 Account Recovery protocol.

SEP-30 is a Stellar Ecosystem Proposal. That means its a proposal for use in applications built on Stellar. SEPs are primarily concerned with interoperability. They provide the foundation for products to interoperate to leverage each other's network effects.

SEP-30 specifically defines an interface that sits between wallets and account recovery servers.

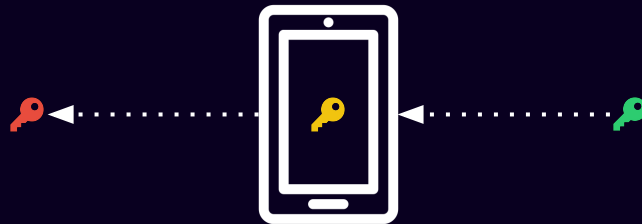
It's an approach to key management that has the goal of making the user experience of a wallet as user friendly as possible and more akin to traditional consumer financial apps that non crypto users would be familiar with.

Its goal is to:

1. Give the user the option of giving up some responsibility.
2. Without giving any single entity control of their account.

**TRANSITION:** A wallet that implements SEP-30 does many of the same things other wallets do, but instead of backing up the master key, it...

# Help the user get a new key.



24

**TRANSITION:** A wallet that is implementing SEP-30 still does many of the same things other wallets do, but instead of backing up the master key, it...

It helps the user get a new key when they need it.

It prevents the user from losing their **account**, and not their key.

The user's Stellar account won't change, its address will stay the same, but the user's signing key changes.



Account  
Registration

POST /accounts/GDBB...

Transaction  
Signing

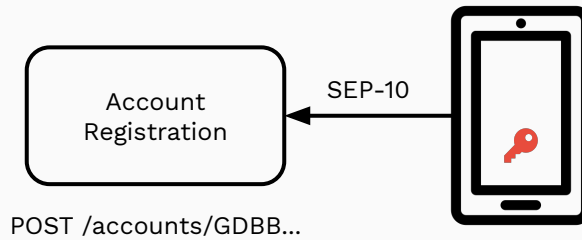
POST /accounts/GDBB.../sign/...

SEP-30 defines an API that a server implements.

And that a client consumes.

That API includes two endpoints:

1. One to register accounts;
2. And, another to sign transactions for that account.



A wallet uses the first endpoint to register an account with the server.

In the request the wallet tells the server what identities are allowed to request signatures for the account.

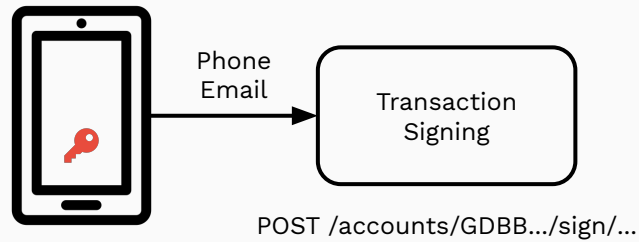
Those identities can be things that can be verified quickly like a phone number, an email address, or login with any OpenID Connect provider.

Or, those identities can be things that require more effort or take more time to verify like a postal address, passport number, driver's license number, or a national identity document.

The wallet authenticates using SEP-10, which is a Stellar Ecosystem Proposal that defines a challenge-response authentication flow that the wallet uses to prove to the server that it possesses keys that can sign for the Stellar account.

This way the server knows that the client should be able to choose who can sign transactions for the wallet.

In the response the wallet gets back a signing address that the server will use to sign transactions for this account. The wallet makes that signing address a signer of the Stellar account.



The second endpoint is transaction signing.

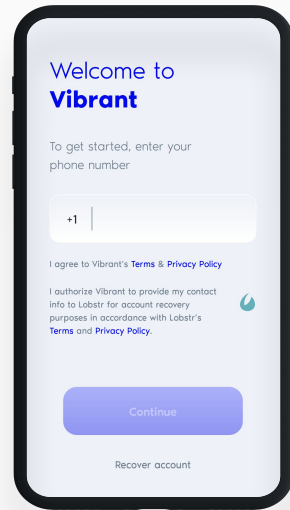
A wallet calls this endpoint with a transaction it wants the server to sign. It'll do this when it has lost its key and needs the server to sign the transaction for it.

The wallet authenticates using an identity provided during registration. Once the server confirms the identity is registered with the account, the server will sign the transaction and return the signature to the wallet.

**TRANSITION:** SEP-30 is currently being used in one wallet...



# Vibrant



28

**TRANSITION:** SEP-30 is currently being used in one wallet...

The Vibrant wallet, which is a non-custodial wallet designed to help users protect themselves from depreciating assets and high inflation.

It supports deposits and withdrawals with anchors, and stores value on the Stellar network.

It uses SEP-30 for account recovery with an independent 3rd party partner, Lobstr.

The Vibrant wallet presents a user experience designed for the end user.

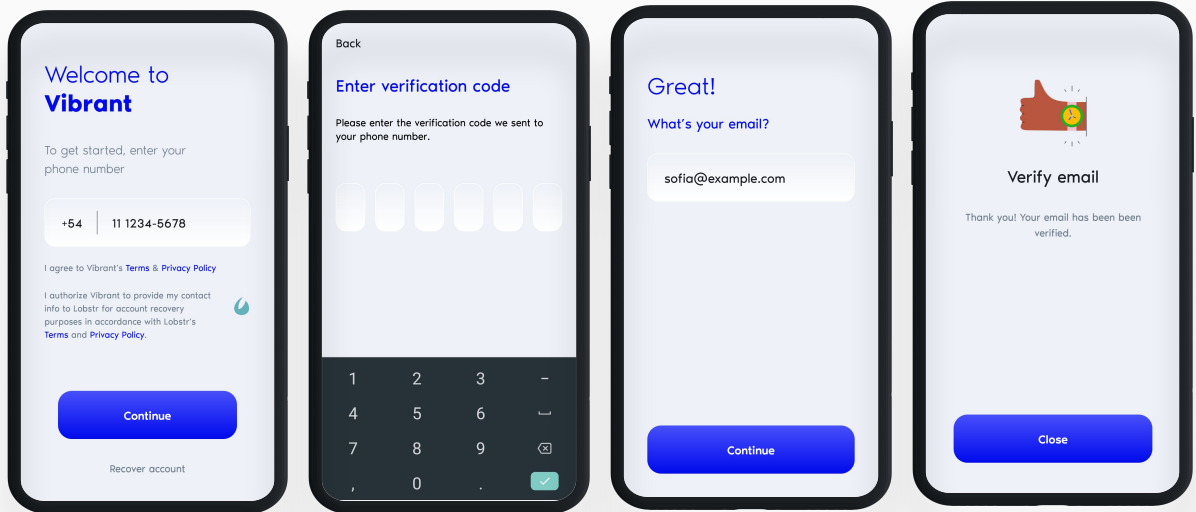
You login with your phone number.

You don't need a password.

You're not required to backup your key, although you can if you want to.

Let's have a look at an example of how Vibrant implements SEP-30 which will show us how a wallet can use the two SEP-30 endpoints to support account recovery.

**TRANSITION:** We'll start with a walk through how Vibrant onboards a new user, then the recovery process.



29

The user experience for onboarding in Vibrant is much like the flow of other end-user applications.

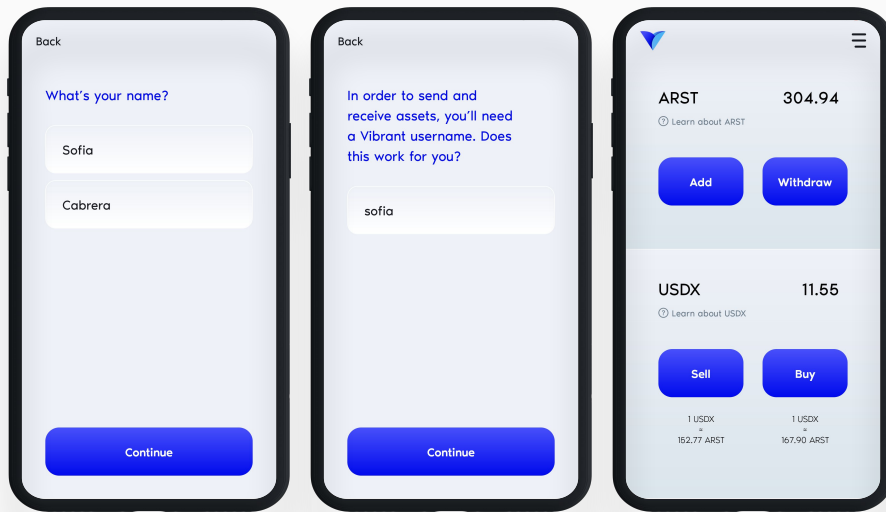
The user is asked for their phone number, and an email address, which will be used as their login credentials.

They verify both by entering a code and clicking a link.

The app uses the phone number and email address to register the account with two independently operated servers that implement SEP-30.

One operated by Vibrant, the other by Lobstr.





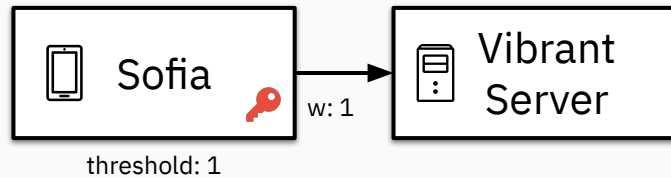
30

The user is also asked for their name and to choose a username.

And that's it, they're in the app.

Everything is setup to support them recovering in the future.

**TRANSITION:** Let's look behind the scenes to see what the wallet did to prepare the user for account recovery...

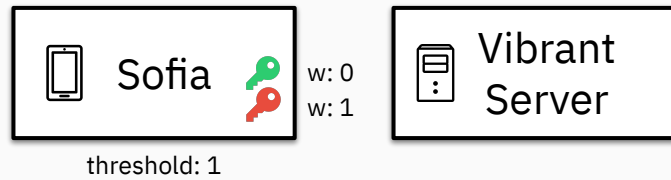


In these diagrams our user is Sofia.

When Sofia creates an account with the wallet, the app on her phone generates a master key and the wallet server creates the account associated with the master key. The master key will be one of those big S secret keys that we looked at at the beginning of the talk and is the red key in the diagram.

Like all accounts on the Stellar network when first created, the master key is the only signer with a weight of 1 and the threshold for signing is 1.

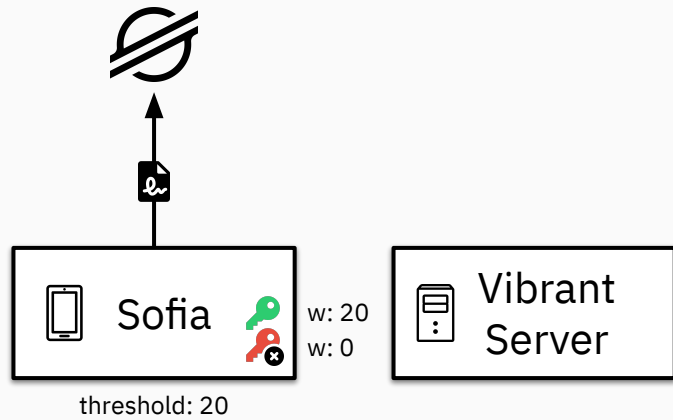
The app on her device doesn't give this key to anyone. It never leaves her device.



Sofia's wallet also generates a second key, what we'll call a device key.

It's a key that is only for this device to use to sign transactions.

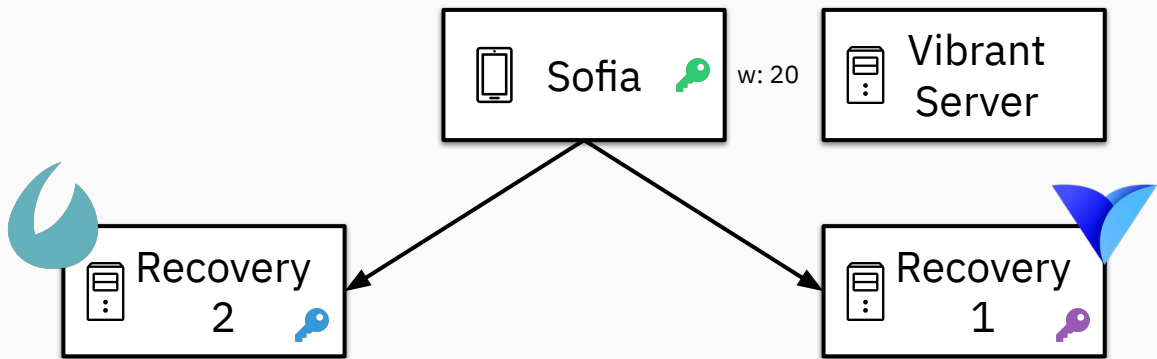
In the same way the master key doesn't leave the device, this new key doesn't leave the device either.



Sofia's wallet submits a transaction to the network making that device key a signer of the account and removing the master key as a signer.

The wallet ups the threshold required to sign transactions to 20, and makes the weight of the new device key 20, and the weight of the master key zero.

The master key now has no control over the account.



The wallet then goes on to register the account with two recovery servers.

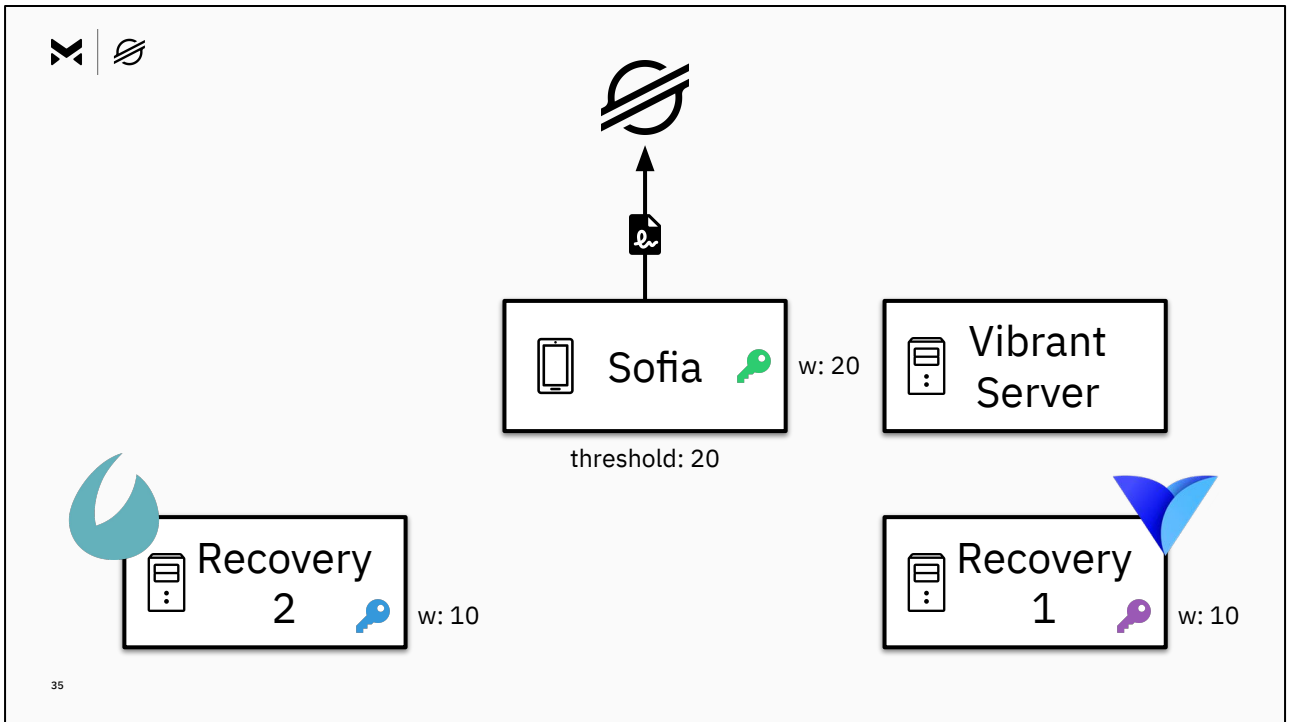
Each server has their own key that they generate themselves.

Each server is operated independently by different entities.

For Vibrant:

- Recovery 1 is operated by Vibrant.
- And, Recovery 2 is independently operated by Lobstr.

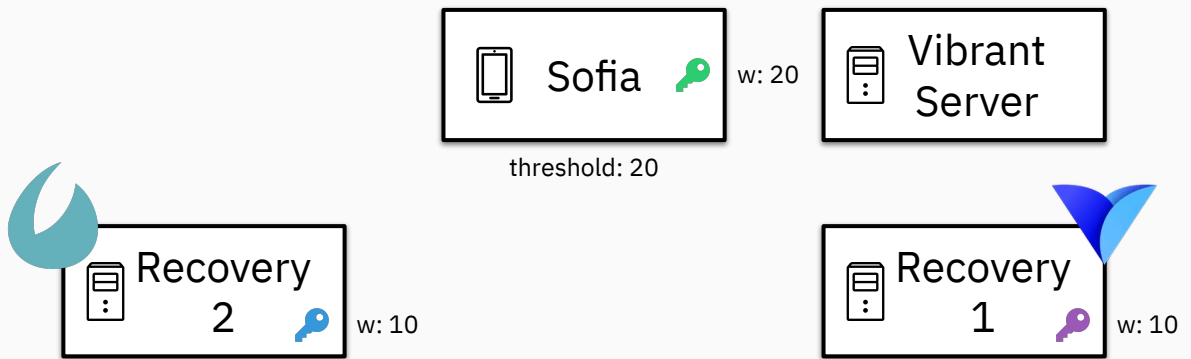
The wallet proves to each server independently that it has authority over the account by signing a SEP-10 transaction, and tells the two recovery servers that anyone who can prove possession of the users phone number or email, should be allowed to request for transactions to be signed.



The wallet submits a transaction to the network adding both recovery servers as signers on the account.

The wallet sets the weight of their signatures so that neither recovery server has independent control over the account.

Both recovery servers signers have weights of 10 and must both sign a transaction to meet the threshold.

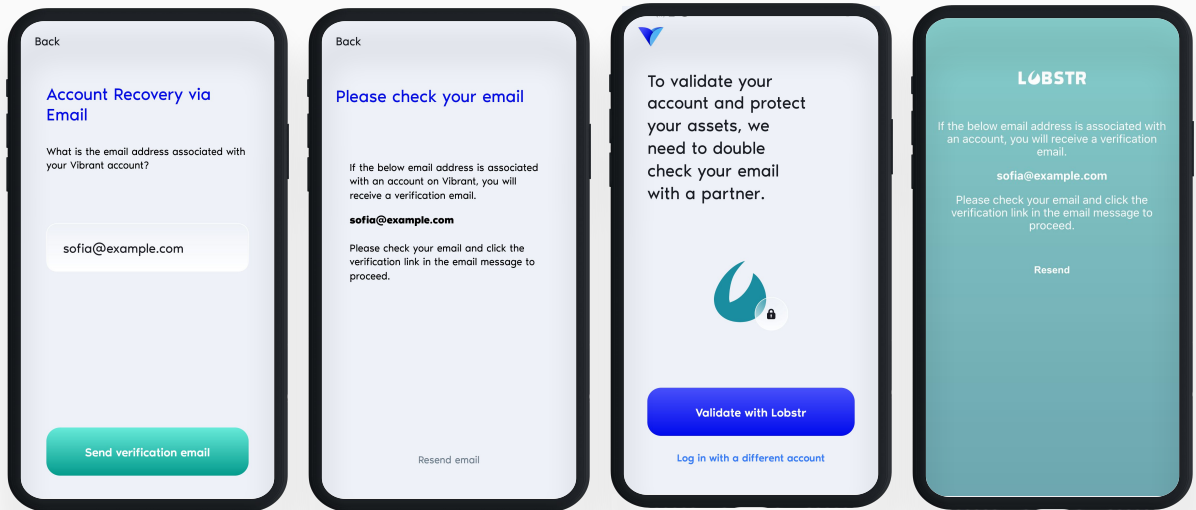


36

Each recovery server will be able to sign a transaction at the request of the user but the transaction will only be authorized if signed by both servers.

The only individual with independent control of the account is Sofia.

**TRANSITION:** Let's walk through the recovery process...let's imagine that Sofia lost her phone and got a new phone.



37

**TRANSITION:** Let's walk through the recovery process...let's imagine that Sofia lost her phone and got a new phone.

She opens the app.

She enters her email address.

She verifies her email address by clicking link that is emailed to her.

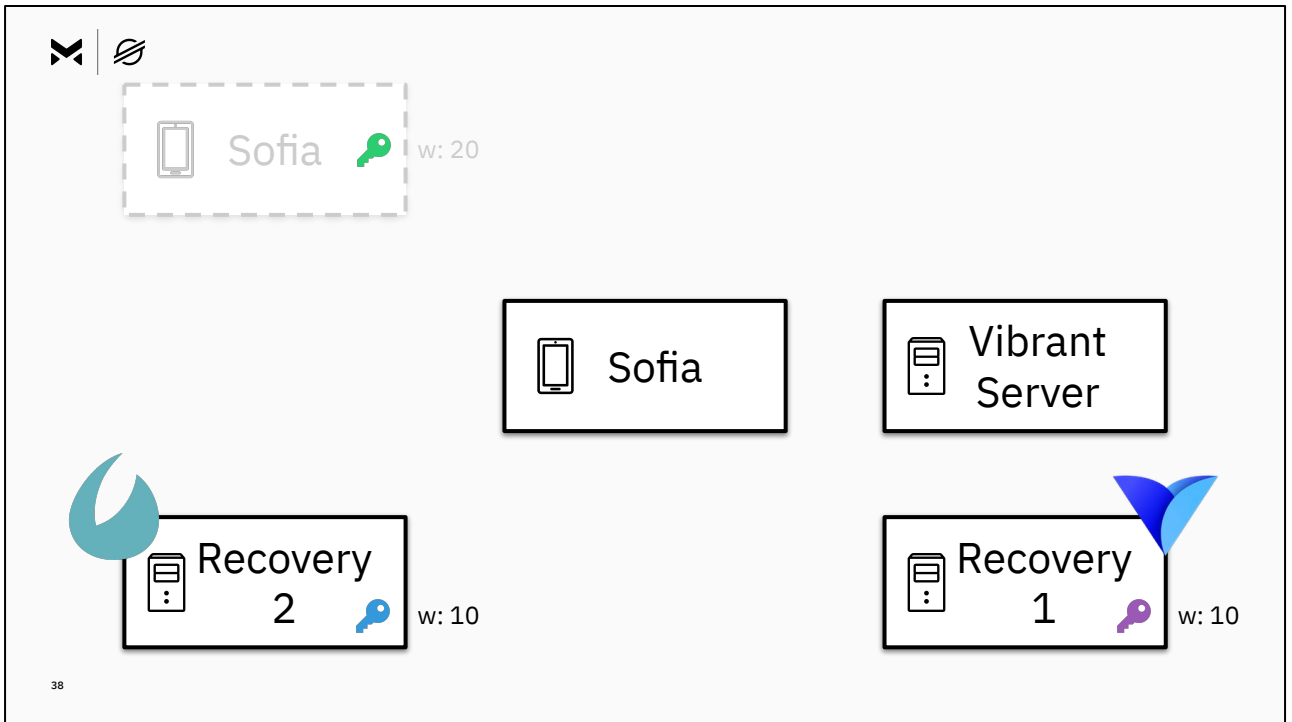
She then verifies her identity with the second recovery partner, in the case of Vibrant, this is Lobstr.

Lobstr verifies independently with a separate verification email.

And that's it, once each verification is complete each recovery server signs the transaction and once both have signed Sofia's device has control of the account.

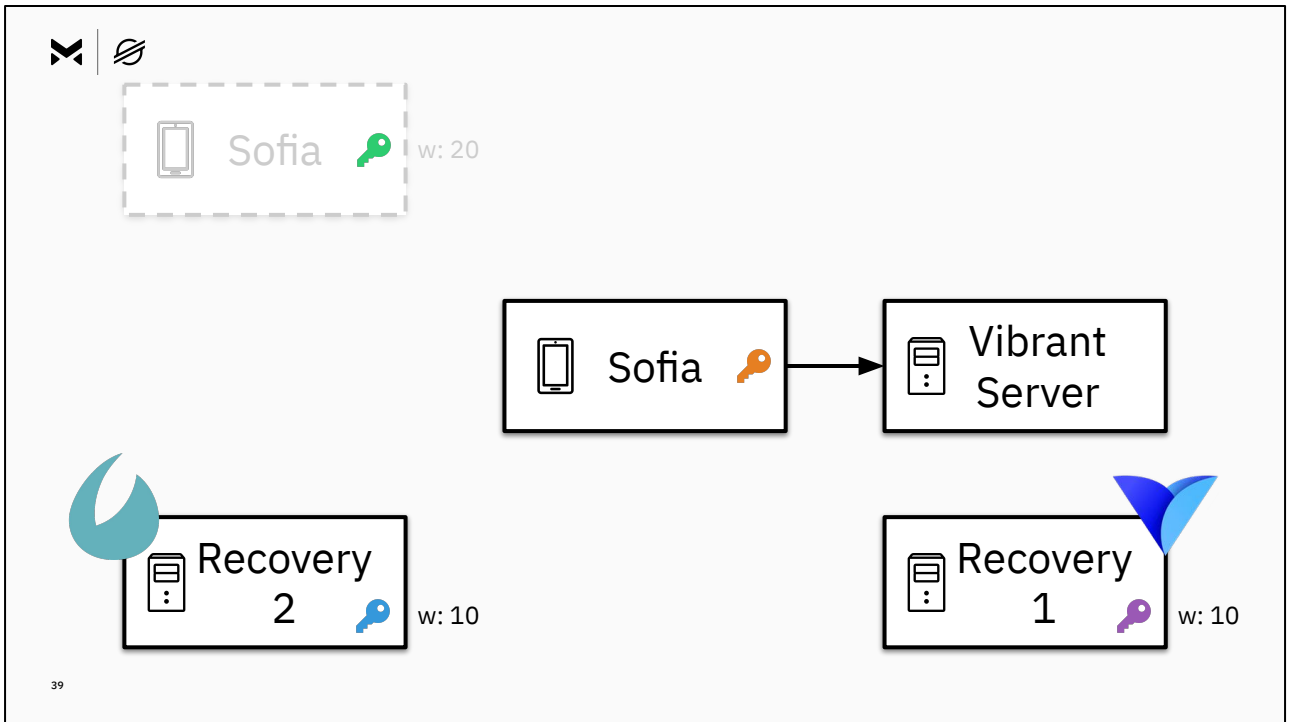
**TRANSITION:** Let's look behind the scenes again to see what the wallet did to recover the account...





Starting from the top, the greyed out phone with the green key is her lost phone that still has her device key on it.

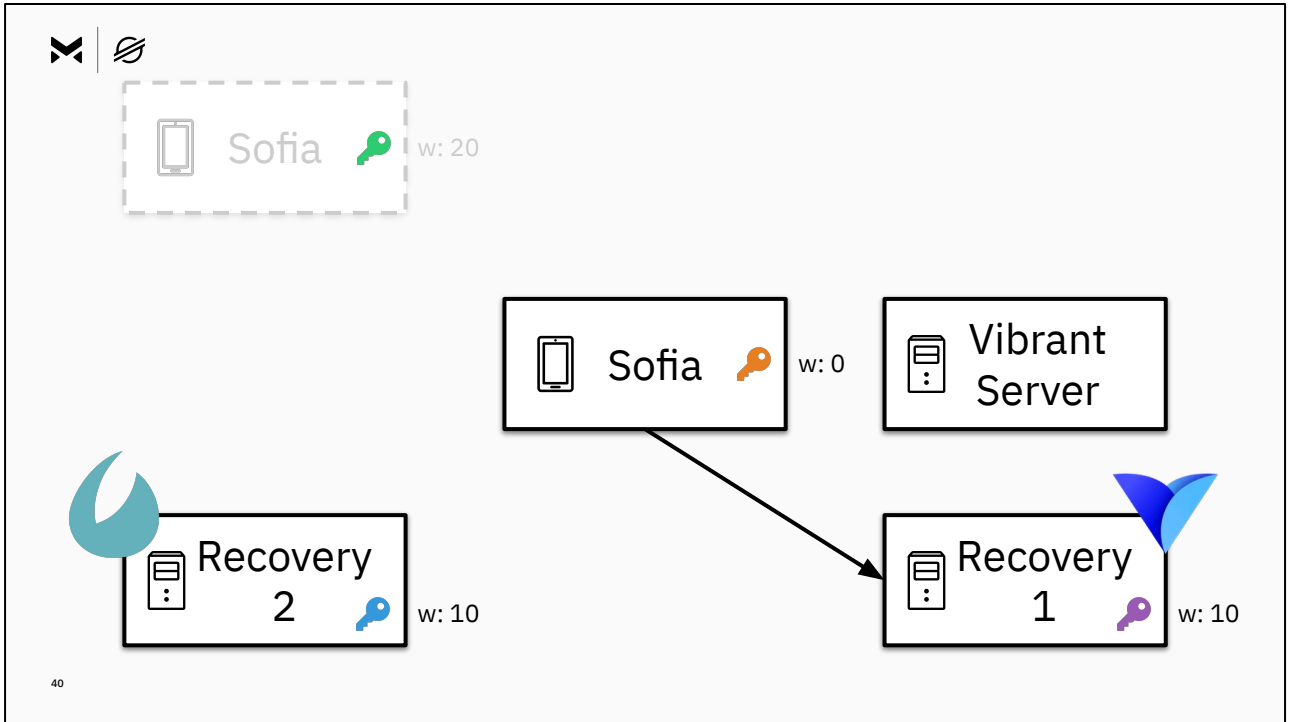
Her new phone in the middle has no key for the account.



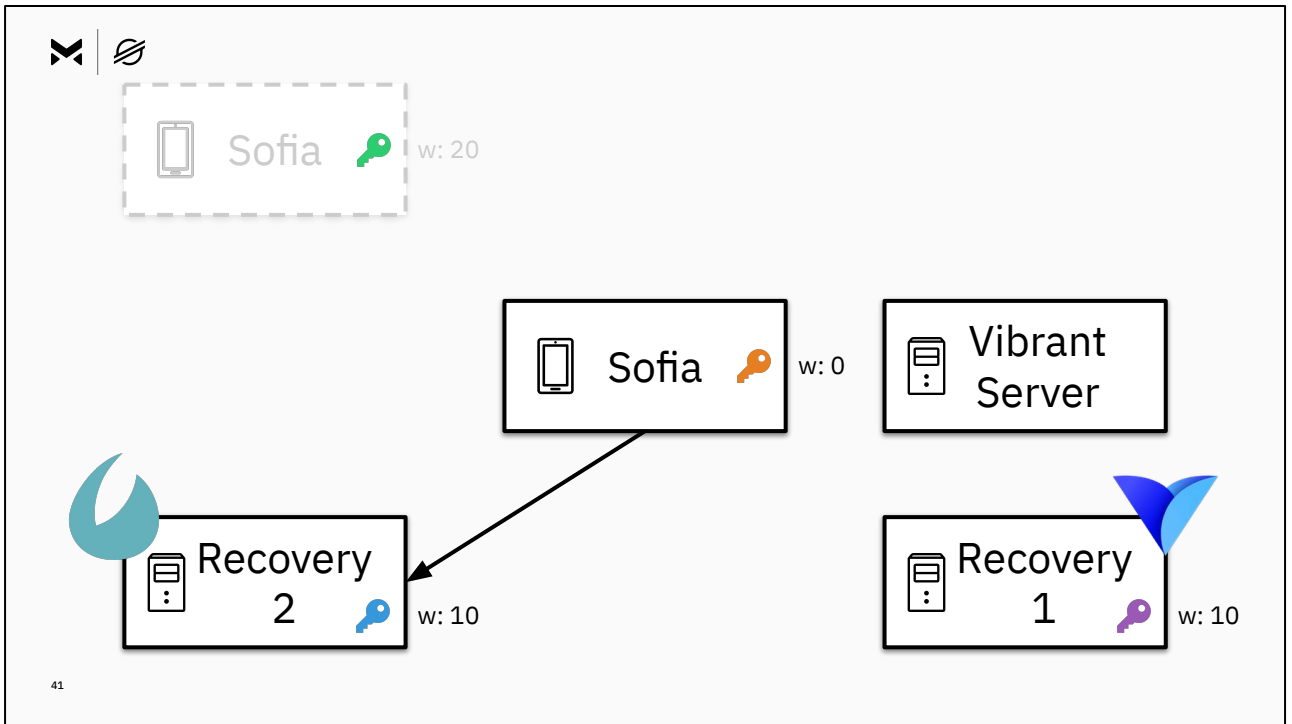
Sofia's new phone goes through the process of collecting and verifying Sofia's identity.

It generates a device key, and signs in to the wallet.

Sofia is signed in, but the new device key is not a signer of her Stellar account, so her device cannot authorize transactions, yet.



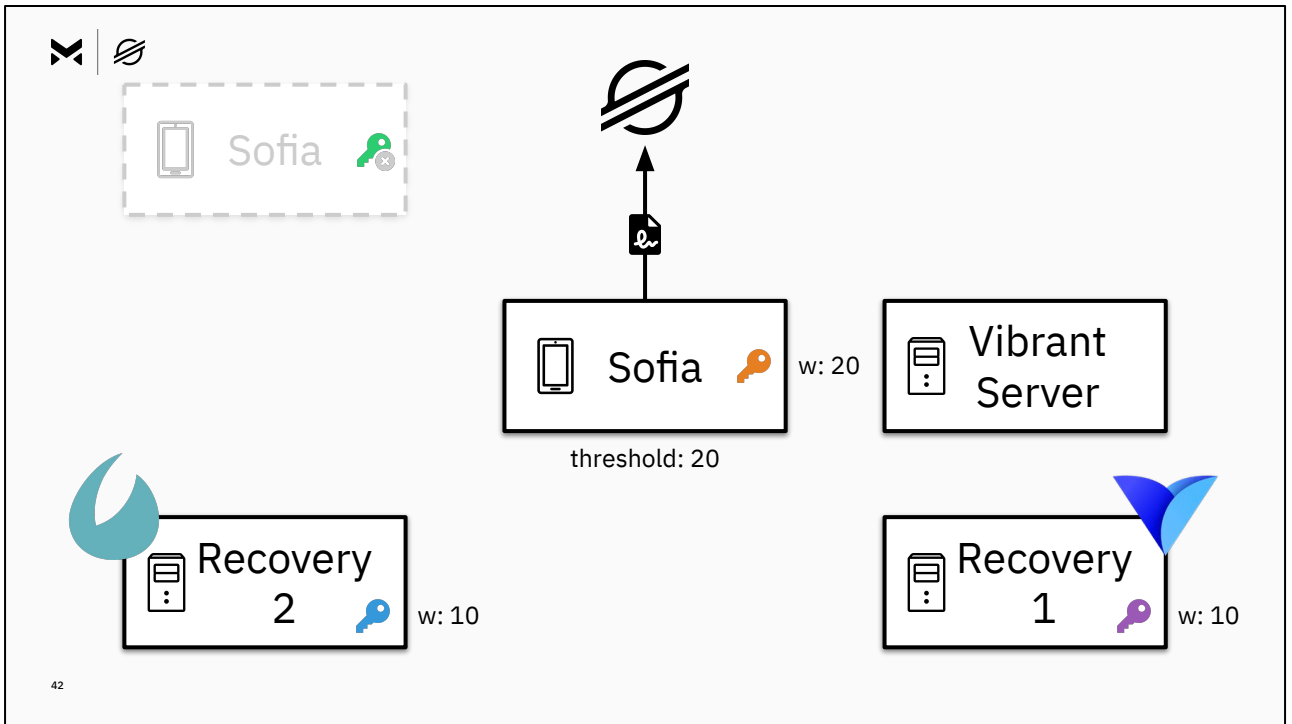
The wallet app talks to the first recovery server and asks the server to sign a transaction that makes her new device key a signer on the account.



Sofia then continues recovery with the second recovery server.

Sofia authenticates with recovery 2 independently because the server is operated independently.

Once authenticated, the server signs the transaction and returns the signature to the wallet.



The transaction is authorized with a weight of 20 and is submitted to the network.

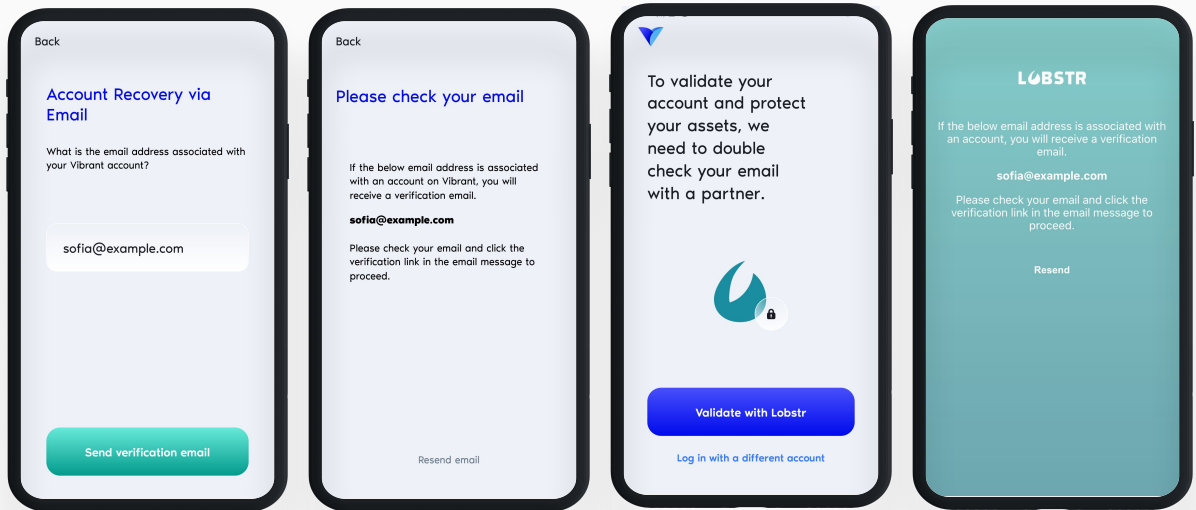
The transaction removes the old signing key that was lost with Sofia's previous phone.

And adds the new signing key that lives on Sofia's new phone.



Sofia is now back in control of her Stellar account.

**TRANSITION:** The steps that Sofia followed and the experience Sofia encountered are similar to...other end-user applications.



44

**TRANSITION:** The steps that Sofia followed and the experience Sofia encountered are similar to...other end-user applications.

## PAUSE

The experience the application creates doesn't bleed into Sofia's life in the form of codes and keys to protect or lose, that are a single point of failure.

## PAUSE

**TRANSITION:** Vibrant is one example of how to implement SEP-30, and if you're implementing SEP-30 you can...choose to give the user more or less responsibility for key management.



**TRANSITION:** Vibrant is one example of how to implement SEP-30, and if you're implementing SEP-30 you can...choose to give the user more or less responsibility for key management.

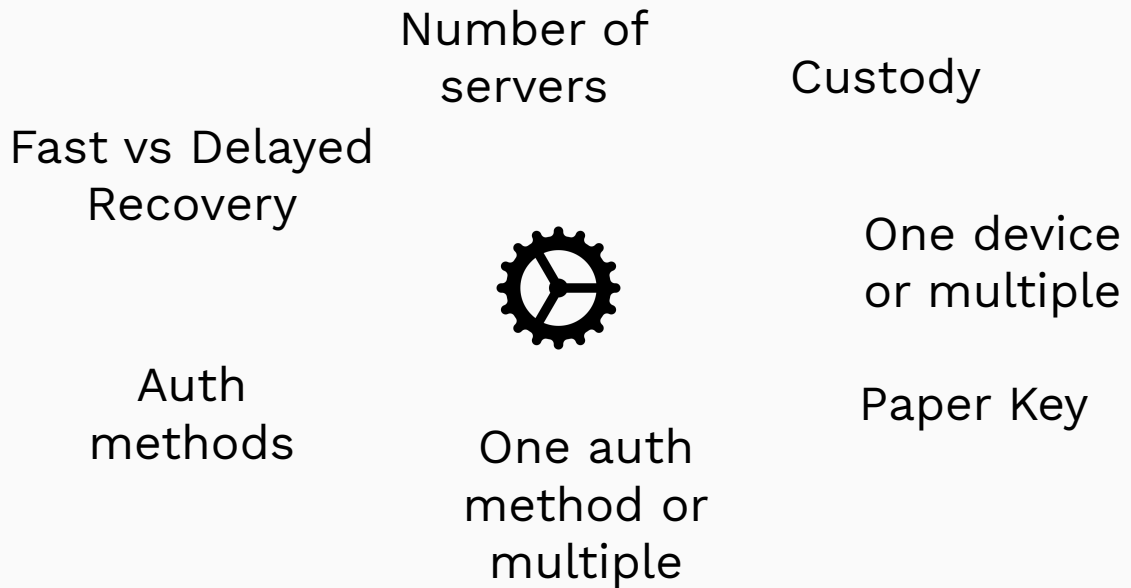
SEP-30 doesn't require that the user:

- Write the key down.
- Store it securely.
- Find it or not lose it.
- Protect it from natural disasters.
- Protect it from thieves.
- Trust a single third party with sole control.

Instead, SEP-30 gives a product developer options and flexibility to choose the responsibilities that make the most sense for their users.

**TRANSITION:** The specification leaves many decisions in the hands of the product implementing it...





46

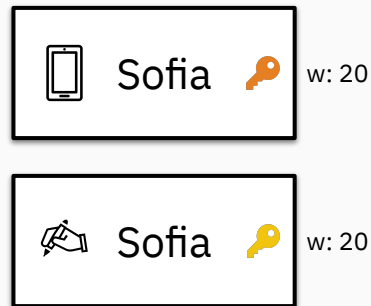
**TRANSITION:** The specification leaves many decisions in the hands of the product implementing it...

The wallet consuming the API chooses which identities to set on accounts at registration, how many servers to register with, and how to divvy up signer weight as well as which transactions to sign at recovery.

The wallet also chooses if SEP-30 is the primary method of account recovery or one of several options.

The server implementing SEP-30 chooses the authentication methods that can be used during recovery, like phone, email, physical address, passport, national ID, etc. Some of these methods will facilitate fast recovery, and others might take time.

**TRANSITION:** Let's have a look at some variations of how SEP-30 could be used...



47

A wallet could implement SEP-30 and offer recovery phrases as an additional key.

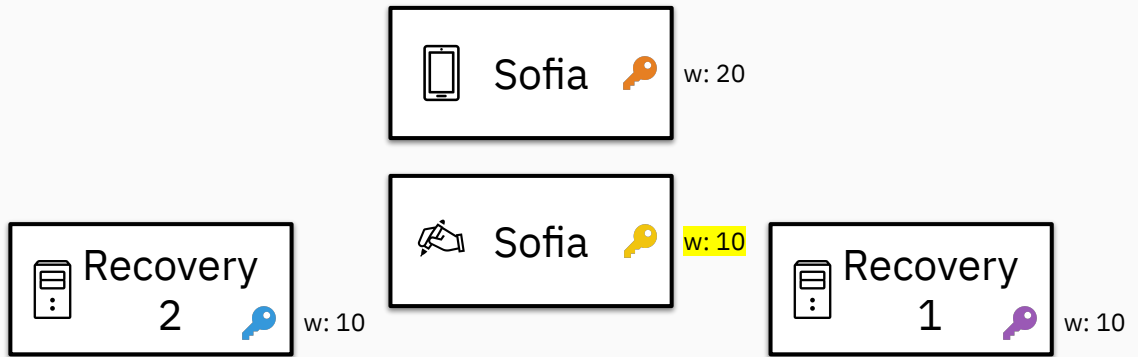
This is actually a feature that Vibrant provides alongside its use of SEP-30.

It gives the user the option to backup their account on their own by getting a recovery phrase.

Unlike other wallets, the recovery phrase is a separate unique key for the account, and not the master key.

This means the key can be revoked independently.

**TRANSITION:** A wallet could configure the recovery phrase to have control of the account on its own, like Vibrant, or...



48

**TRANSITION:** A wallet could configure the recovery phrase to have control of the account on its own, like Vibrant, or...

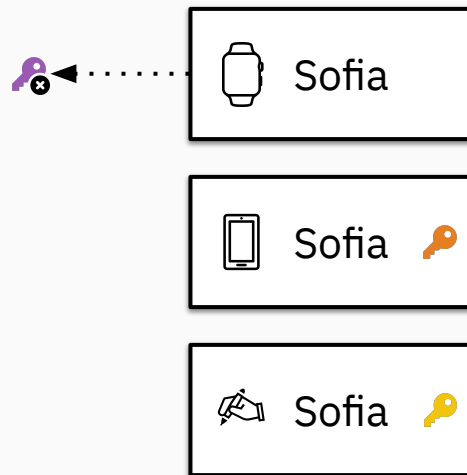
In conjunction with a recovery server.

In this case signatures are required from both recovery servers, or one recovery server and the recovery phrase.



A wallet could support multiple devices where each device generates its own key.

**TRANSITION:** A wallet could deal with lost devices by...revoking individual keys.



**TRANSITION:** A wallet could deal with lost devices by...revoking individual keys.

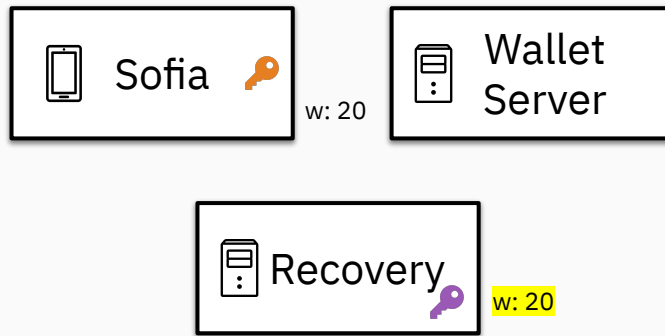
A lost watch containing a wallet could be quickly revoked by removing its key as a signer

Without interrupting the use of other devices.

Because each device has its own signing key, the source of transactions submitted to the network could be auditable.

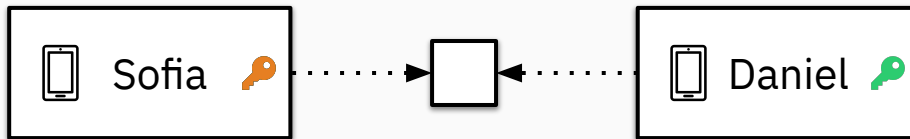
If a user saw a transaction that was unfamiliar, they could identify which device signed for it, and revoke that device immediately.

Or it might simply help them identify how the transaction happened.



A wallet could use SEP-30 to shift key management to a third party who provides key management as a service.

A wallet could do this by giving a single SEP-30 server signing control of the account.



SEP-30 can also support other use cases of account recovery. A wallet could support sending assets to people who are not yet users.

SEP-30 makes it possible for a user to create a temporary account that another user can recover to claim the assets.

For example, if Sofia registers the temporary account with the recovery servers not only with her own phone number, but with Daniel's too.

Daniel could download a wallet, register and recover the temporary account that Sofia created, in order to claim the assets.

A wallet could use this feature to facilitate invitations of new users.

**TRANSITION:** SEP-30 is a foundation of many possibilities...



# SEP-30 Account Recovery

[stellar.org/protocol/sep-30](https://stellar.org/protocol/sep-30)

53

**TRANSITION:** SEP-30 is a foundation of many possibilities...

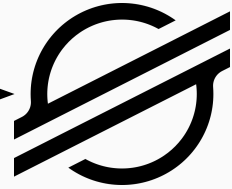
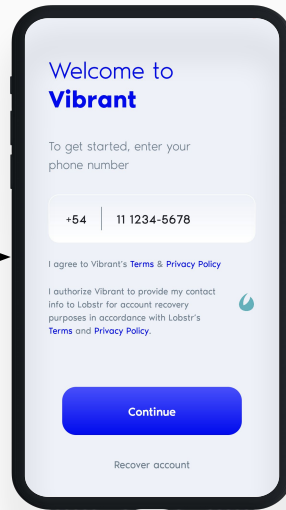
SEP-30 gives wallets the option to create an experience comparable to consumer financial products.

While avoiding trust in single third party with sole control.

With the flexibility to be tuned to different business and product needs.

**TRANSITION:** SEP-30 gives wallets the opportunity to...create great user experiences and great user experiences drive adoption.





**TRANSITION:** It gives wallets the opportunity to...create great user experiences and great user experiences drive adoption.



# Getting Started



[stellar.org/protocol/sep-30](https://stellar.org/protocol/sep-30)



[stellar.org/blog/the-importance-of-key-management-recovery](https://stellar.org/blog/the-importance-of-key-management-recovery)



[stellar.public #key-management](#) on Keybase



[groups.google.com/g/stellar-dev/c/SFr2dHBZlsY](https://groups.google.com/g/stellar-dev/c/SFr2dHBZlsY)

55

Thank you for listening.

If you'd like to read the SEP-30 specification it's a 15 minute read.

We've also discussed more of the why behind SEP-30 on the SDF blog.

If you have questions I'll be in the Meridian Virtual Lounge at 12:30 PST.

I also hang out in the key-management channel on Keybase.

And there's a thread on the stellar-dev mailing list about SEP-30 if email is more your style.

As more products implement SEP-30 we expect them to push the boundaries of how it is defined today.

And, we're eager to hear of challenges and improvements.

So, contributions and feedback are much appreciated!

Thank you!