Stellar

User-Friendly Key Management with **SEP-30 Recoverysigner**

Leigh McCulloch
Software Engineer
Stellar Development Foundation

Hi everyone,

Welcome to the Stellar Development Foundations Engineering talk series.

My name is Leigh McCulloch and I'm a software engineer at the 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 Stellar Ecosystem Proposal 30.
SEP-30 is an approach to make key management user-friendly in products built on Stellar.

User-Friendly Key Management is maybe a bold statement to make because key management is rarely user friendly.

But that's what we're talking about today.

Stellar

Table of Contents

User-Friendly Key Management with SEP-30 Recoverysigner

First I'm going to set some context and touch on:

- What key management looks like on the Stellar network
- How key management relates to wallets
- And common approaches that wallets take in the Stellar ecosystem and in the blockchain space in general

Then we'll jump into:

- The approach taken by SEP-30
- We'll walk through an example step-by-step of how SEP-30 works
- And then reflect on why this approach is valuable

We'll also have a Q&A at the end, so if you have questions during the talk, drop them into the YouTube chat box.

Stellar

Key Management

So, what is key management and why do we care.

At the core of owning a Stellar account, or an account on many blockchains, is being in possession of a key.

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

Stellar

GDBB4A5CNKPXZOHN DKFH5RLY7MHJMBP7 5FPLR2Y4F63OMDHA3 XBAKUQ3

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

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 an example of what the public key looks like on the Stellar network. It starts with a G.

The public key is your account address.

People send funds to this address and they will show up in your account balance.

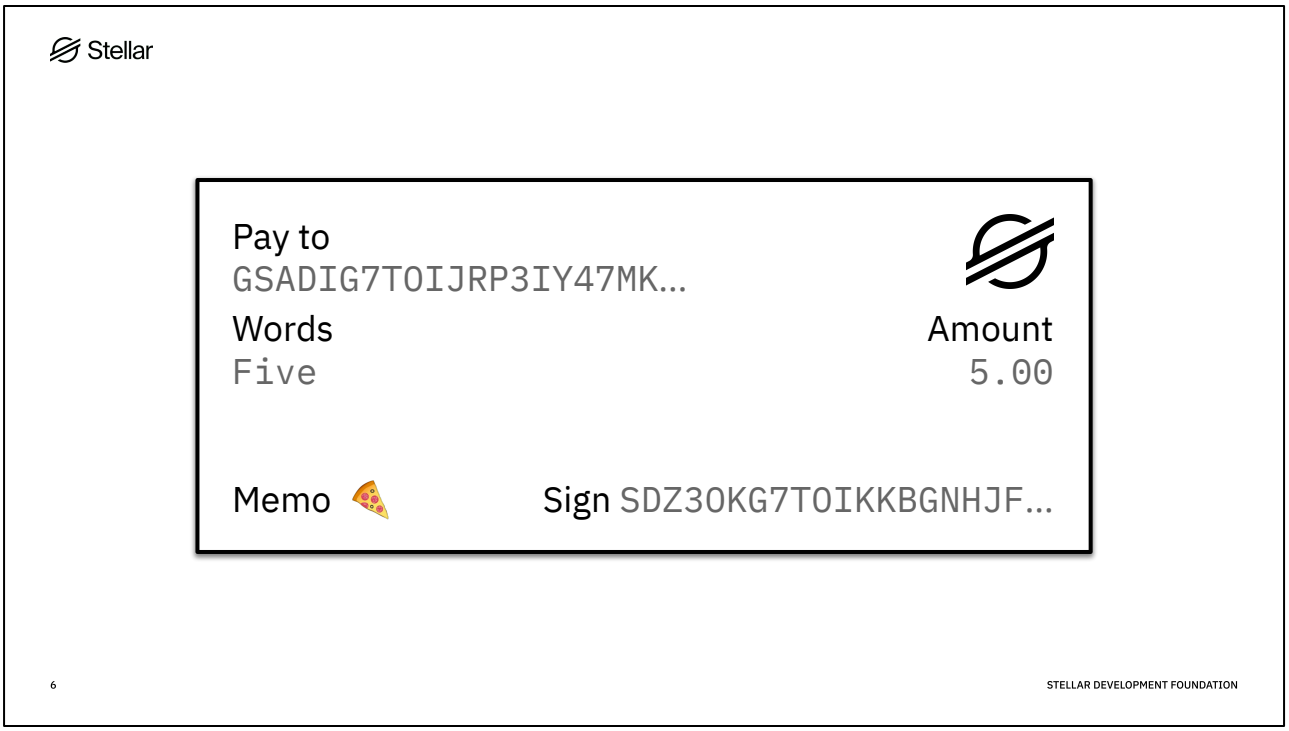
This is what the secret key looks like. It starts with an S.

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 not change its configuration, your master key is what you use to sign transactions.

It's what you use to control your account.

TRANSITION: So what do you we mean by sign transactions...



So, Stellar accounts have signers.

A signer is a key that can authorize transactions for a Stellar account.

And the master key is the default **signer** of the account.

Being a signer of a Stellar account is sort of like being a signer on a bank account. You can sign checks for that bank account, and banks will accept the check as valid.

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.

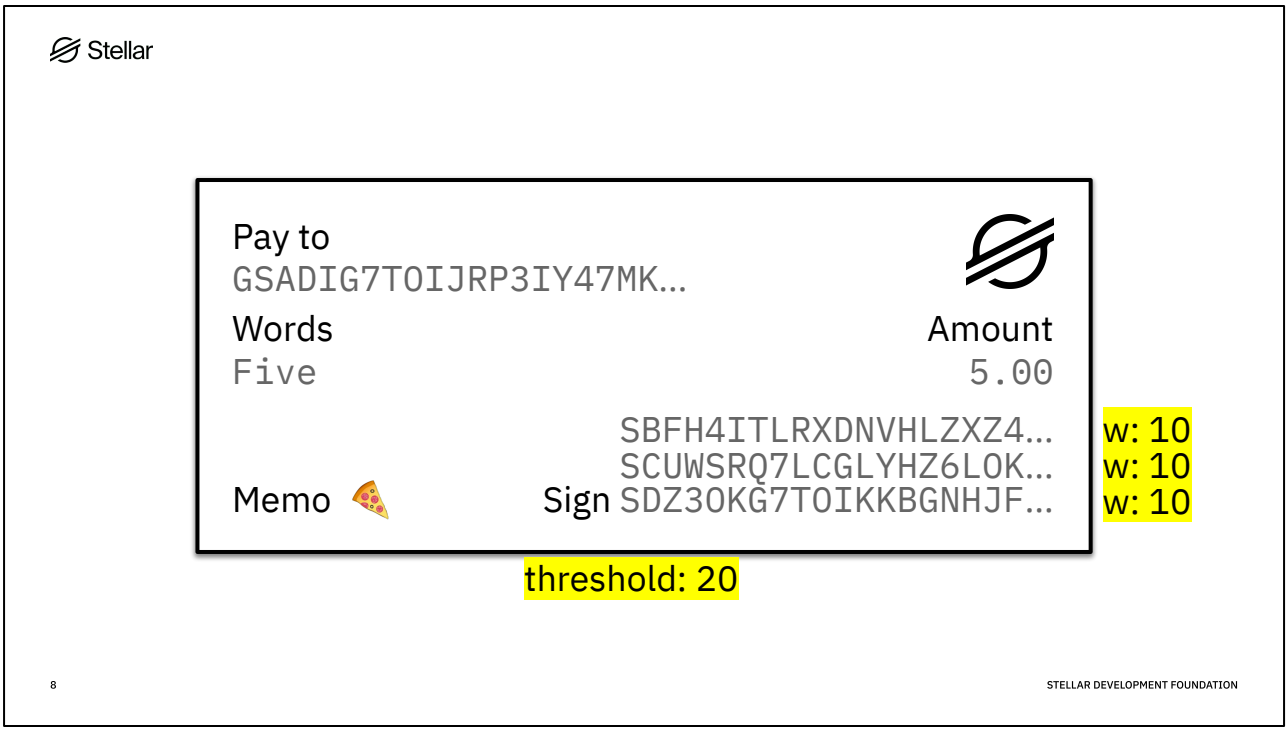
TRANSITION: Accounts can also have more signers...in addition to or instead of the master key.

A signer is a signing key, just like a 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.

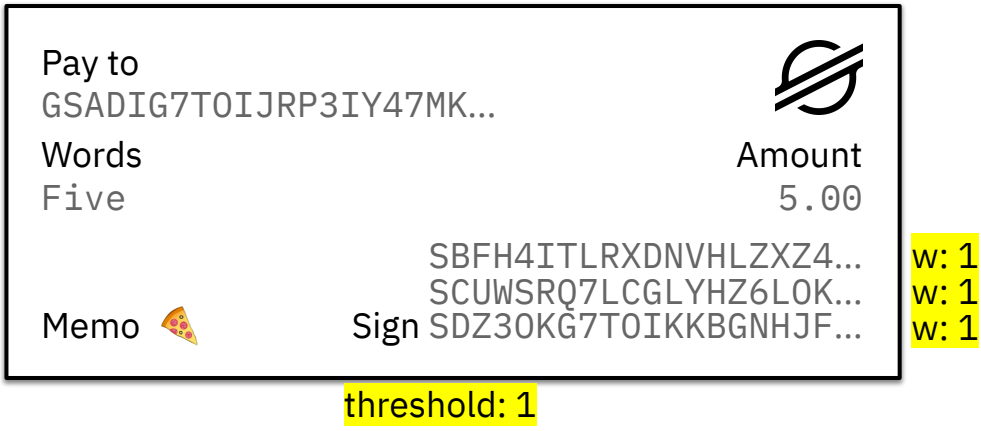
TRANSITION: An example of this is...



This account has a threshold of 20 that has to be met, and so at least two of these signers with weights of 10 need to sign the check for it to be valid.

TRANSITION: Another example could be...

Stellar



This account that has a threshold of 1, where any signer could sign the check by itself for the check to be valid, like a basic joint bank account.

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

But, most accounts on the network only use the master key, and only have one signer. Some wallets have used an additional signer as a 2FA, but control of the account still hinges on possession of the master key.

Key management is how we manage all these keys.

TRANSITION: Wallets play a pivotal role in key management because...at the core of a wallet is key management.

Stellar

Sign Transactions

Store

Protect

Prevent Loss

TRANSITION: Wallet applications play a pivotal role in key management because...at the core of a wallet is key management.

A wallets core job is to hold your Stellar account, and so its core job is to:

- Store your key.
- Protect your key.
- Help you use your key, sign transactions.
- Prevent you from losing your **key**.

TRANSITION: Let's talk about...some common approaches to key management.

Stellar

Common Approaches

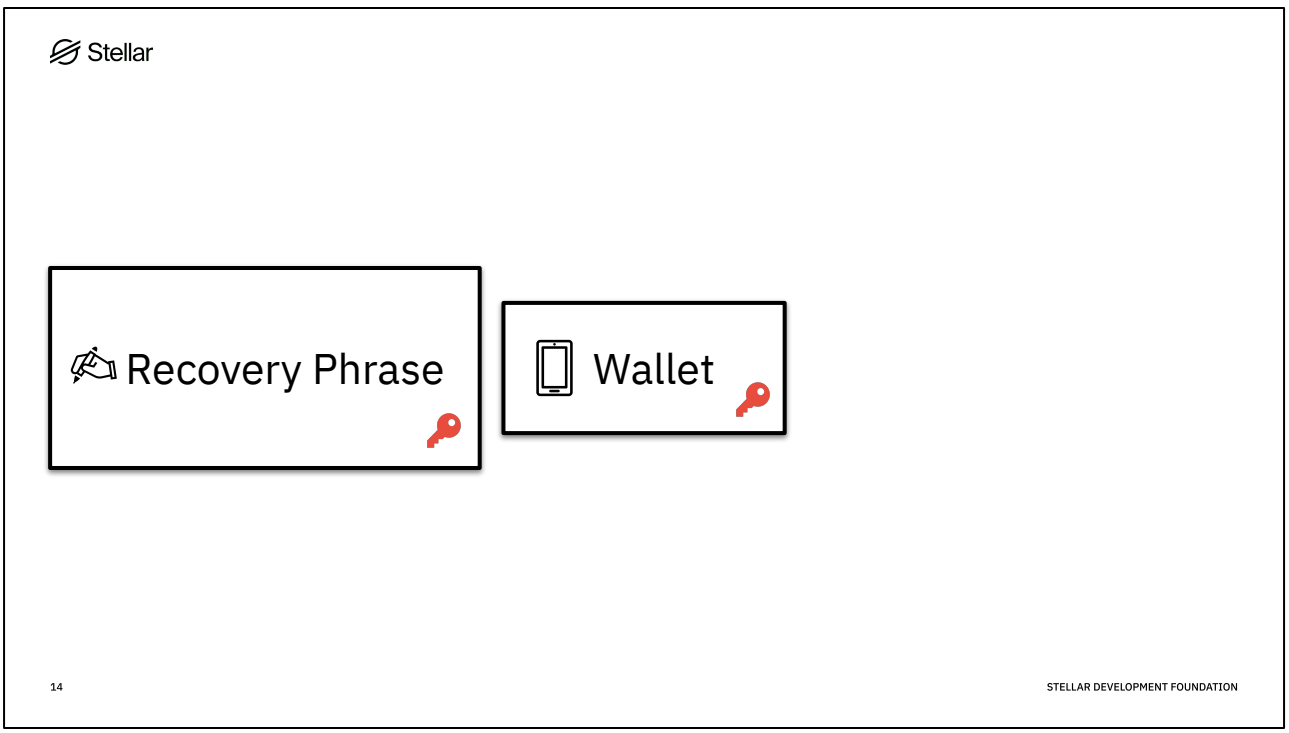
Let’s talk about some common approaches to key management.

Stellar

Many wallets:

- Only use the master key. This means your account has one key and one signer.
- Store your key on the device.

They then help you not lose your key using some different methods:



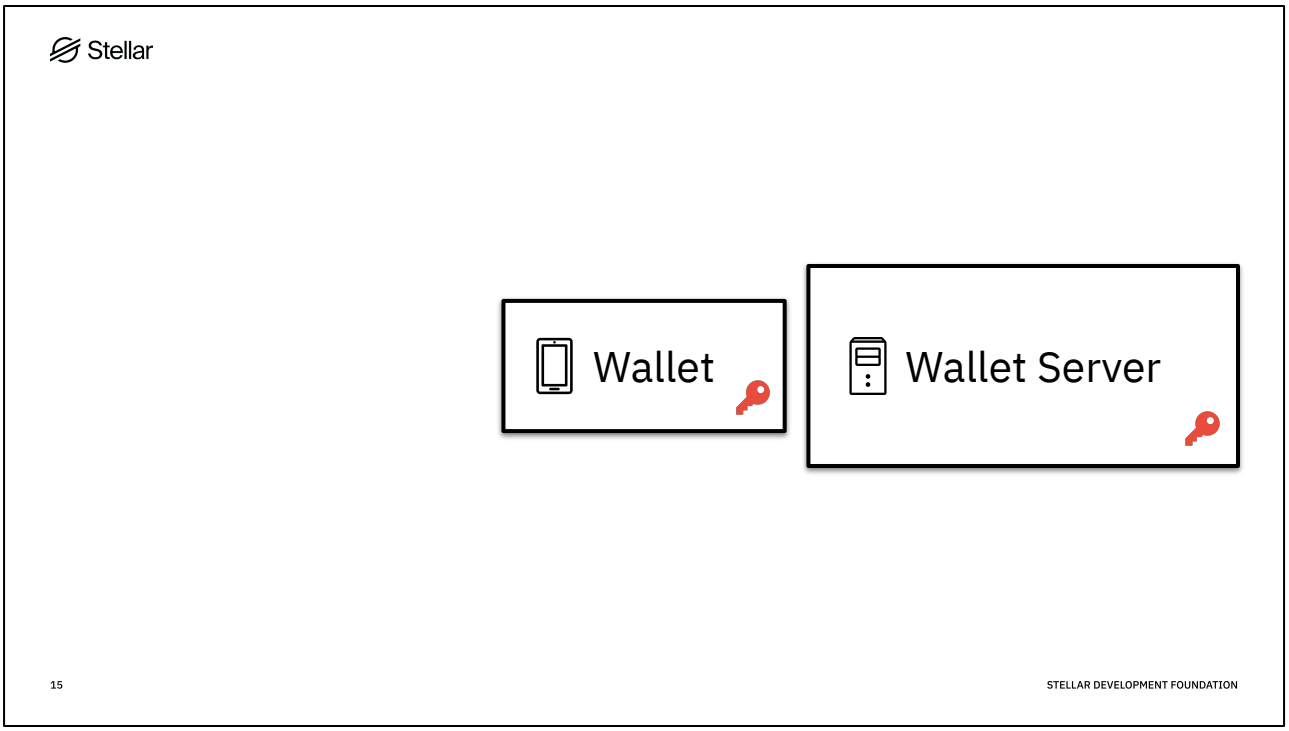
One approach is to give you the key in some form to write down or print out.

The key lives in two places:

- In your wallet on your phone.
- And on paper as a backup if you lose your phone.

This is great in terms of simplicity. Straightforward to implement. Your key is backed up, but it presents some challenges.

1. It puts a lot of responsibility on the user.
2. The user needs to write down the key exactly correct.
3. If they lose the paper it's written on, they've lost their backup.
4. If someone else finds the paper, their account is easily stolen.
5. It can be a jarring experience to ask a user to write down a lot of text immediately after installing a new app.



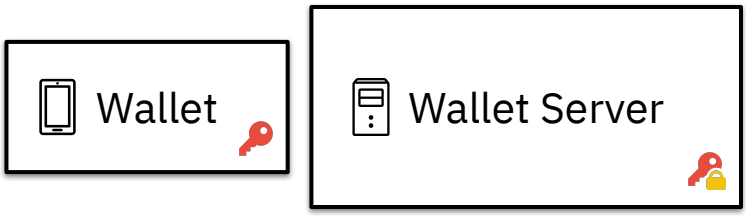
Another approach is to store a copy of the key on the Wallet's server.

But this also has some challenges.

The user must trust the wallet server completely because the server will have access to the key and have control of their account.

TRANSITION: Some wallets address these challenges by...

Stellar



TRANSITION: Some wallets address this challenge by…

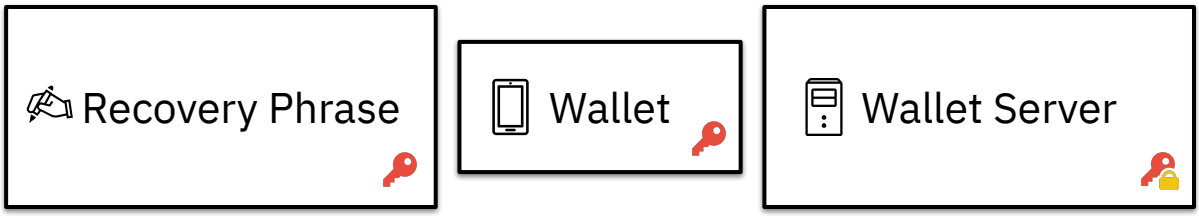
Encrypting the key using a password that the user enters into the wallet but is never shared with the wallet server.

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

But this only works for wallets that require the user to make a password that won't be shared with the wallet server which is yet another password for the user to lose.

Stellar

Most wallets manage a master key...



...and backup the master key

So all wallets do key management, and most focus on managing one key, the **master key**.

And backing up the master key.

Stellar

SEP-30 Recoverysigner
A User-Friendly Approach

Let’s talk about the approach SEP-30 takes.

SEP-30 Recoverysigner is a Stellar Ecosystem Proposal. That means its a proposal that contains new standard or a change to existing standard that is built on top of the Stellar network and for use in the Stellar ecosystem.

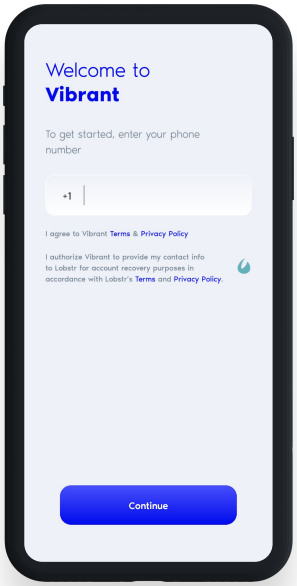
SEP-30 is an approach to key management that’s focus is making key management, and specifically the preventing loss part, as user-friendly as any other consumer application. Its goal is to:

1. Work without recovery phrases.
2. Have no server with sole control of your account.
3. User does not need to remember a password to encrypt keys.

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

Stellar

Vibrant



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

The Vibrant wallet, which is a non-custodial US dollar savings wallet that stores value on the Stellar network.

The Vibrant wallet is currently in open beta, and presents a user experience designed for the general public.

You login with your phone number.
You don't need a password.

TRANSITION: A wallet like Vibrant that is implementing SEP-30 still does many of the same things other wallets do like...

Stellar

Sign Transactions

Store

Protect

Prevent Loss

A wallet implementing SEP-30 still does many of the same things other wallets do like:

- Stores a key.
- Protects a key.
- Help you use your key, sign transactions.
- Prevents loss

But the loss it prevents is different. it prevents you from losing your **account**, and not your key.

A wallet implementing SEP-30 won't manage and backup a single master key.

TRANSITION: Instead, the wallet will...

Stellar

Manage device keys and...

...help the user get new keys

TRANSITION: Instead the wallet will...

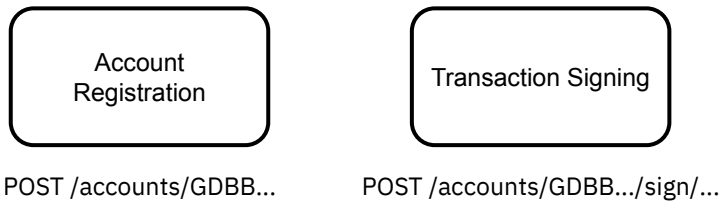
Manage device keys and help the user get new keys.

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

TRANSITION: Let's have a look at what SEP-30 is...

Stellar

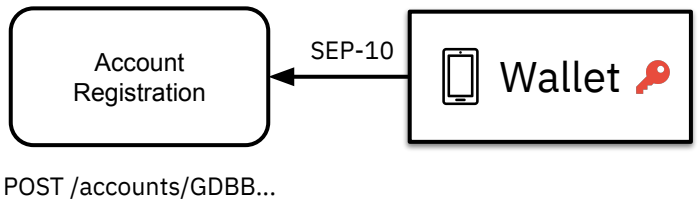
SEP-30 API Endpoints



SEP-30 defines an API that provides two endpoints, one to register accounts and another to sign transactions for that account.

Stellar

SEP-30 API Endpoints



POST /accounts/GDBB...

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 like a phone number or an email address.

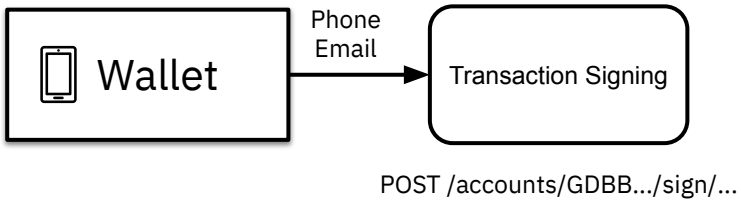
The wallet authenticates using SEP-10, which is a Stellar Ecosystem Proposal that defines a challenge-response authentication flow that the wallet uses to prove to the server that it possesses keys that can sign for the Stellar account.

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

In the response the wallet gets back a signing address for the server and it makes that address a signer of the Stellar account.

Stellar

SEP-30 API Endpoints



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

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. If the phone or email matches an identity stored with the account the server will sign the transaction for the Stellar account and return the signature to the wallet.

TRANSITION: Let's have a look at an example...and walk through how a wallet can use these two endpoints to support account recovery.

Stellar

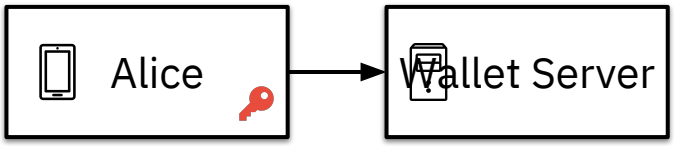
Example

Recovering a Stellar Account

TRANSITION: Let’s have a look at an example...and walk through how a wallet can use these two endpoints to support account recovery.

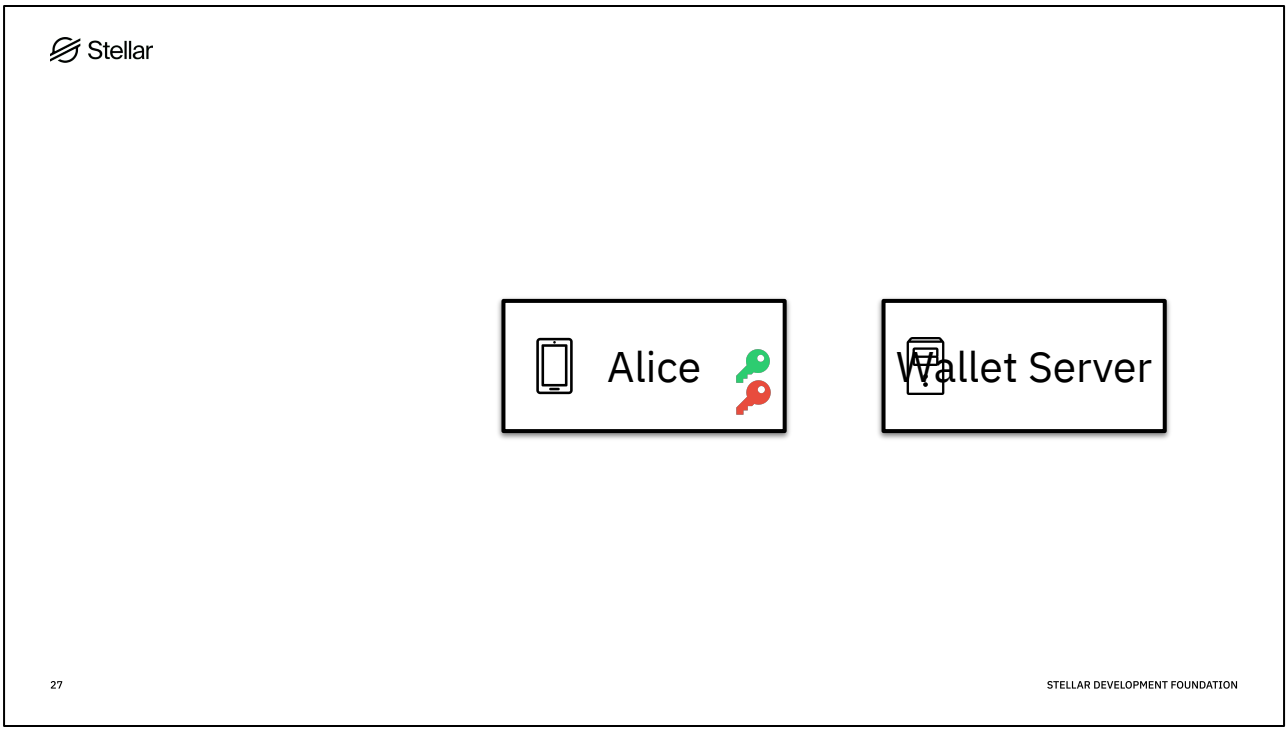
For the example we’ll give the user who is using the wallet a name, and call her Alice.

Stellar



When Alice 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 addresses that we looked at at the beginning of the talk and is the red key in the diagram. The red key never leaves her device.

The app on her device doesn't give this key to anyone. With SEP-30 we can implement the wallet so that there is no need for this master key to be backed up anywhere.

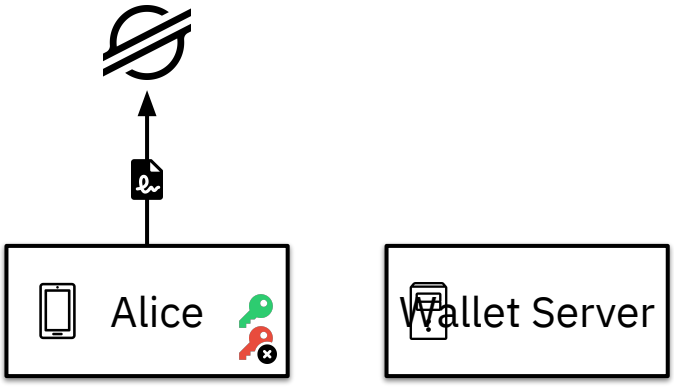


Alice's wallet also generates a second key, what we'll call a device key.

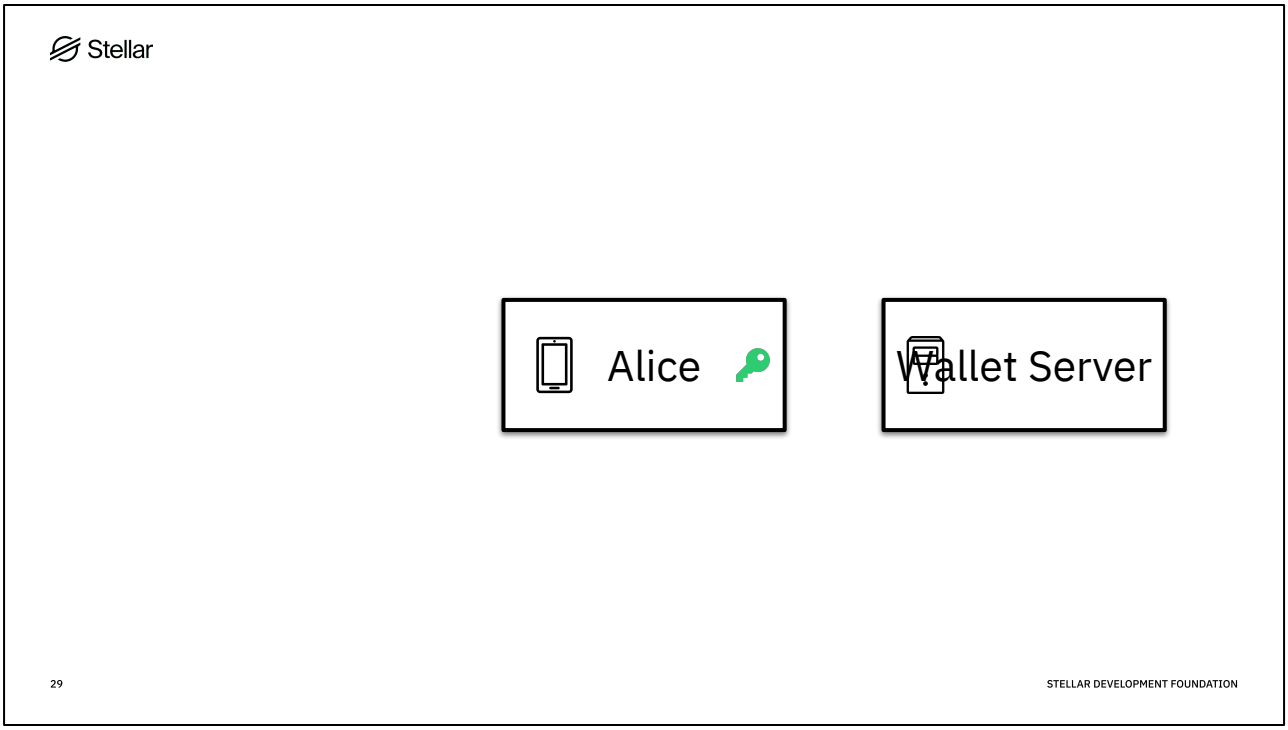
It's a key that is only for this device to use to sign transactions.

In the same way the master key is not shared with anyone, the device key is not shared with anyone either.

Stellar

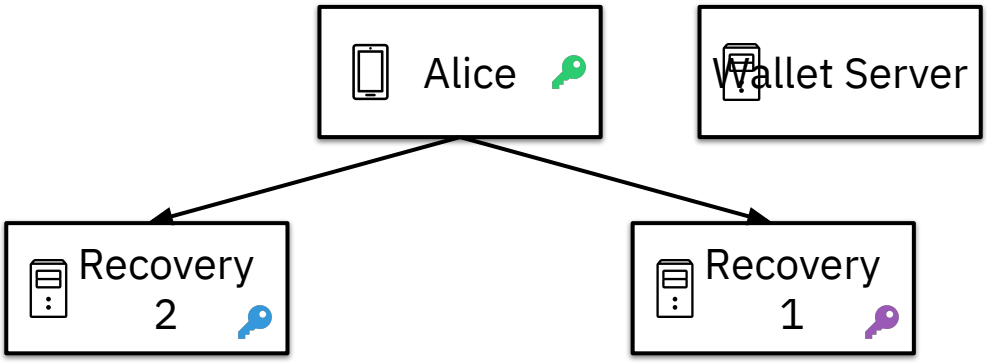


Alice's wallet submits a transaction to the network making that device key a signer of the account and removing the master key as a signer.



And then deletes the master key, it’s not useful anymore.

Stellar

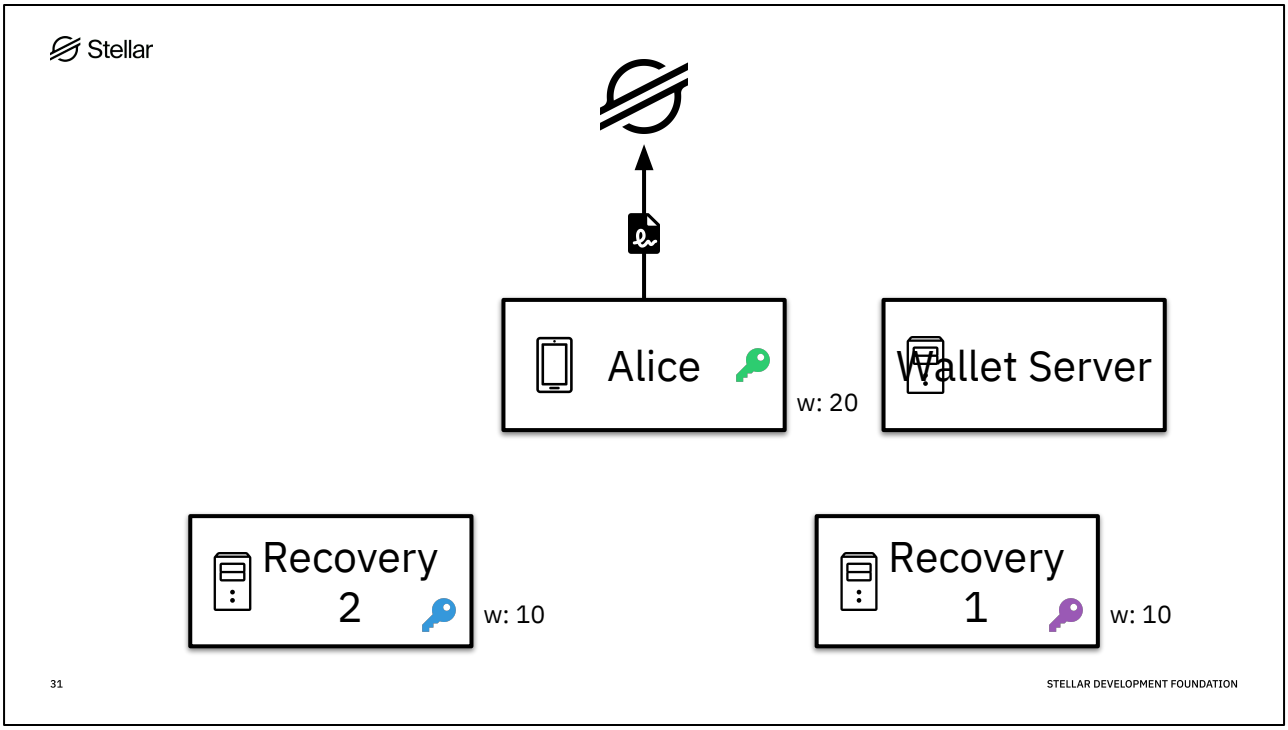


The wallet then goes on to register the account with two recovery servers.

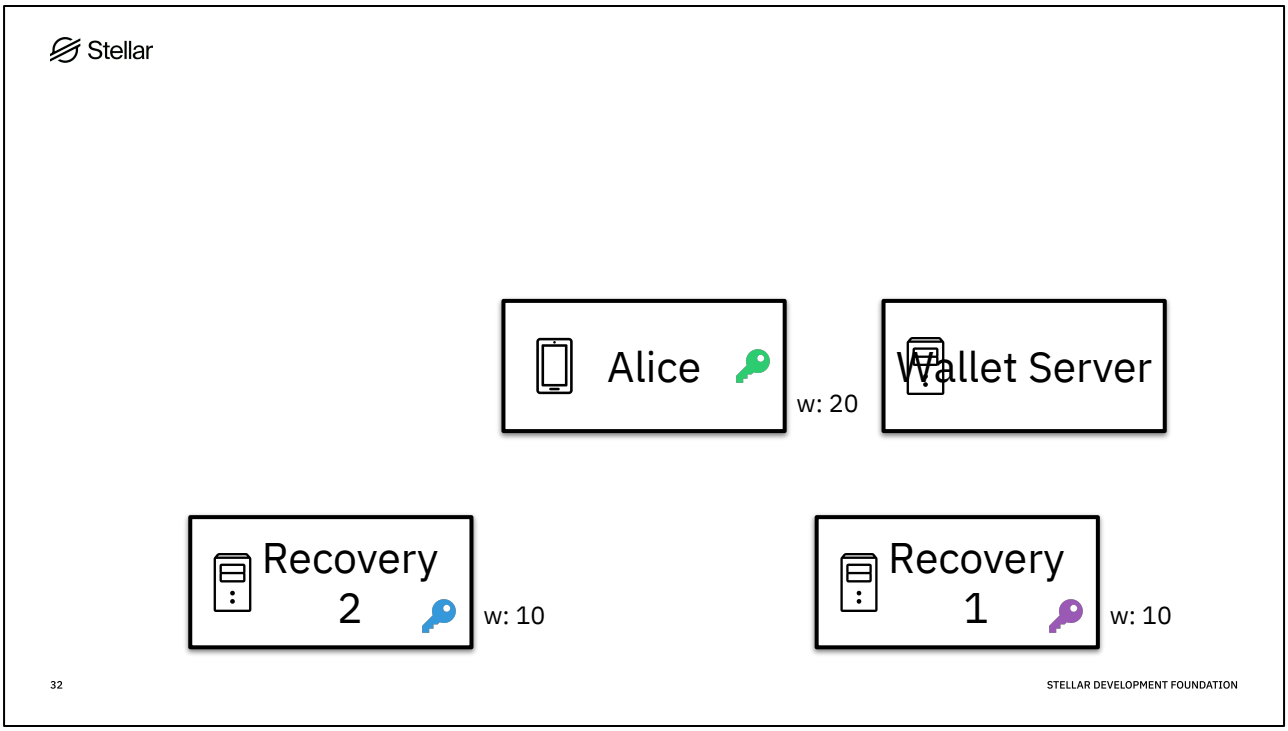
Each server has their own key that they generate themselves.

Each server is hosted and controlled independently by different entities.

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.



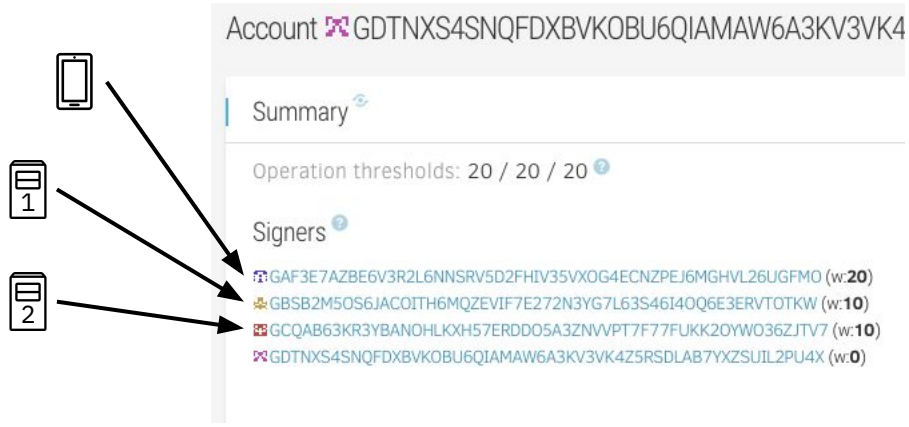
The wallet submits a transaction to the network adding both recovery servers as signers on the account, but limiting the weight of their signatures so that neither recovery server has independent control over the value in the account.



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 another independent party.

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

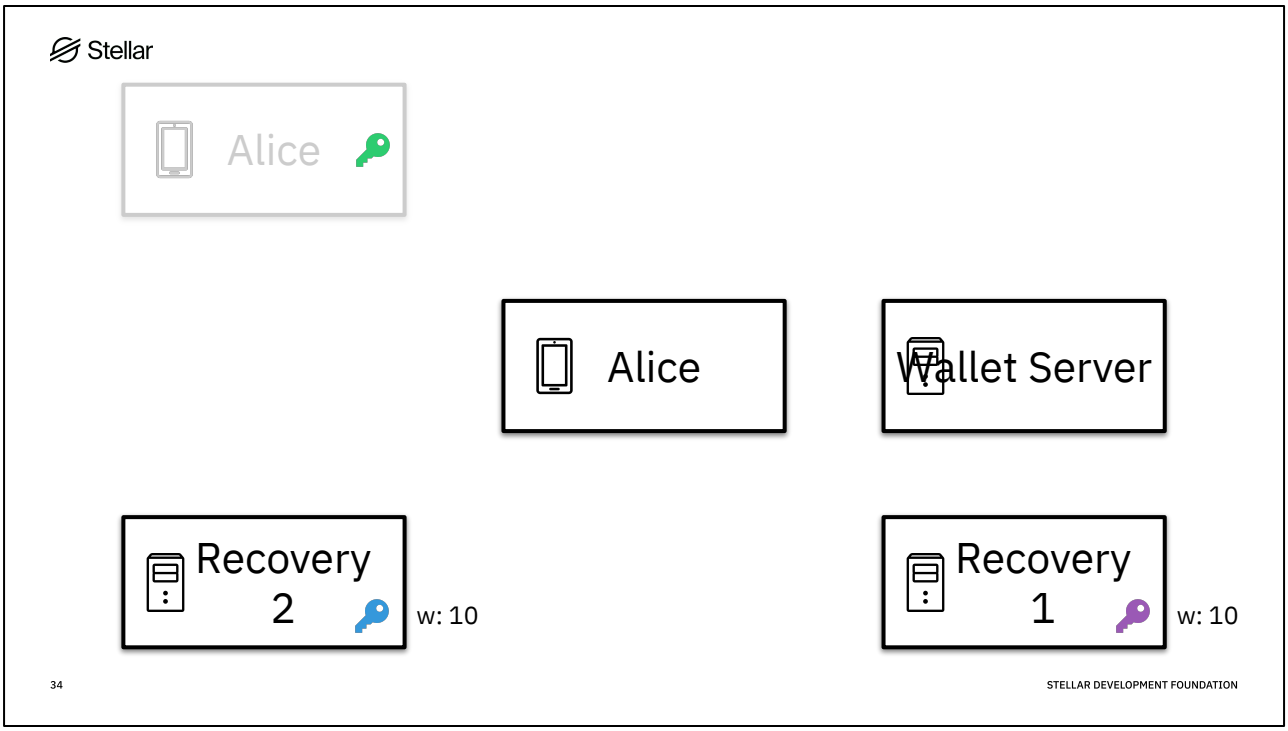
Stellar



Here's an example of what it looks like on a public block explorer.

This is an account on the public network with:

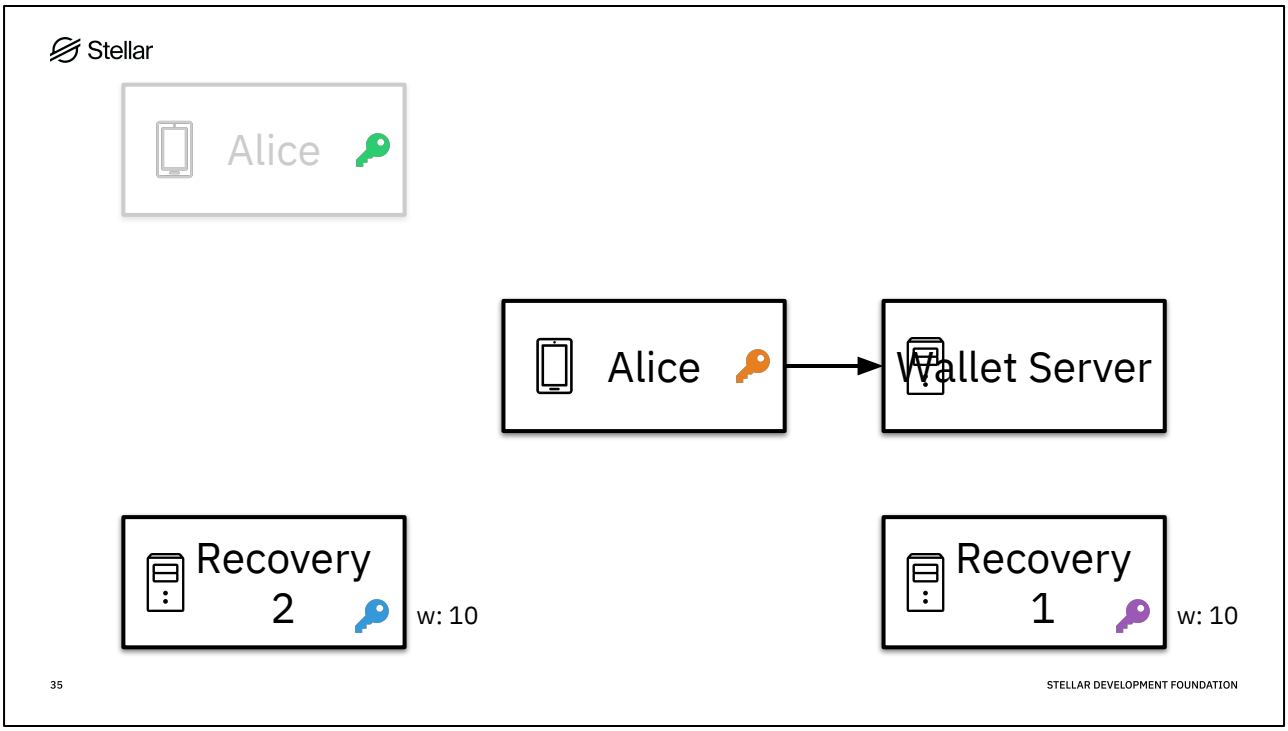
- Its thresholds set to 20, requiring signatures from signer's that add up to 20.
- There are four signers:
  - The master key at the bottom with a weight of zero meaning it has no control over the account.
  - A device key with a weight of 20 so that the device has control of the account.
  - Two recovery servers that each have a weight of 10.



Let’s imagine that Alice lost her phone and got a new phone.

The greyed out phone in the top-left is her lost phone that still has her device key on it.

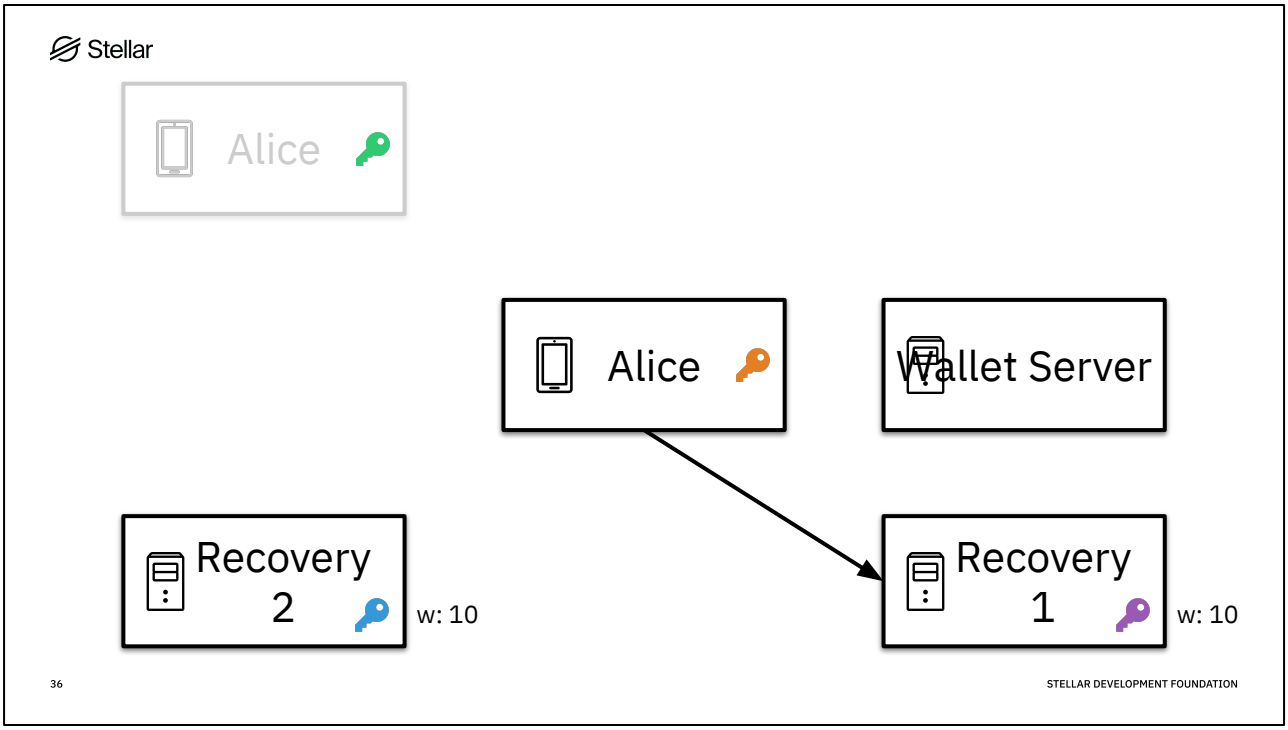
And her new phone has no key for the account.



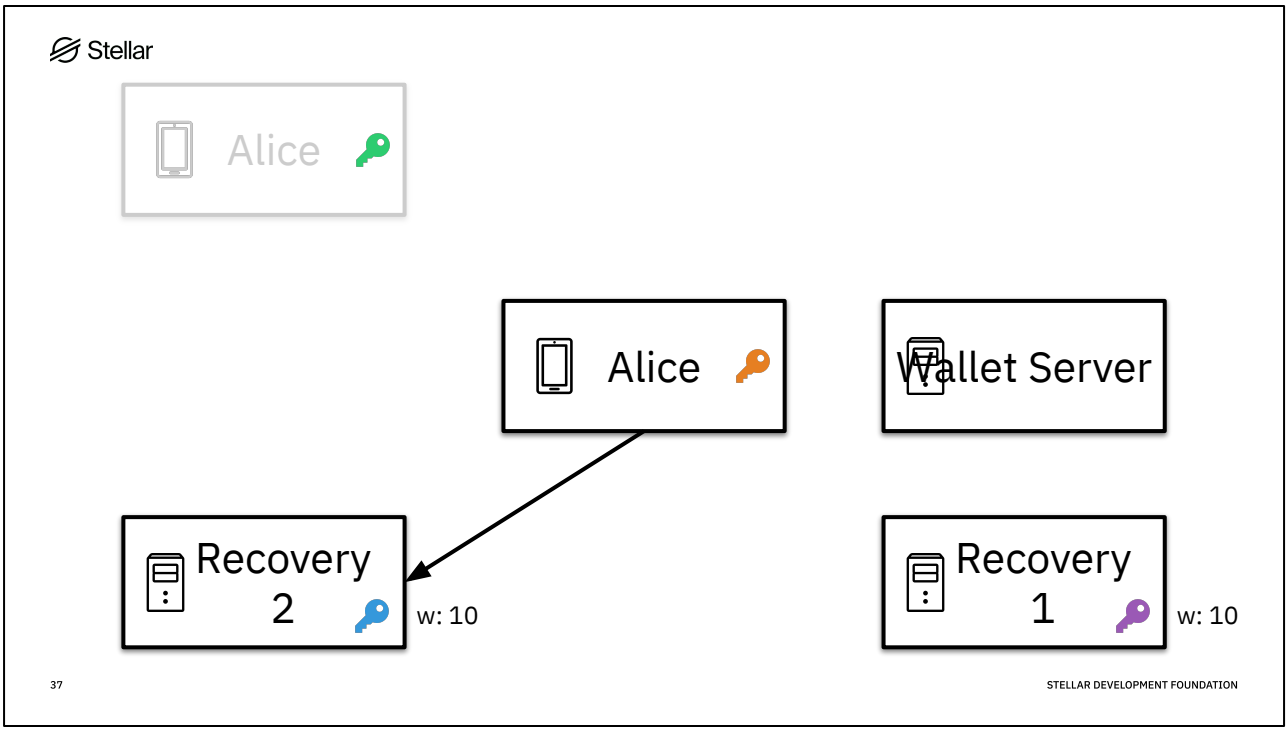
Alice's new phone goes through the same process that her previous phone did when it signed up with the wallet server.

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

But the new device key is not a signer of her Stellar account, so her device cannot authorize transactions, yet.



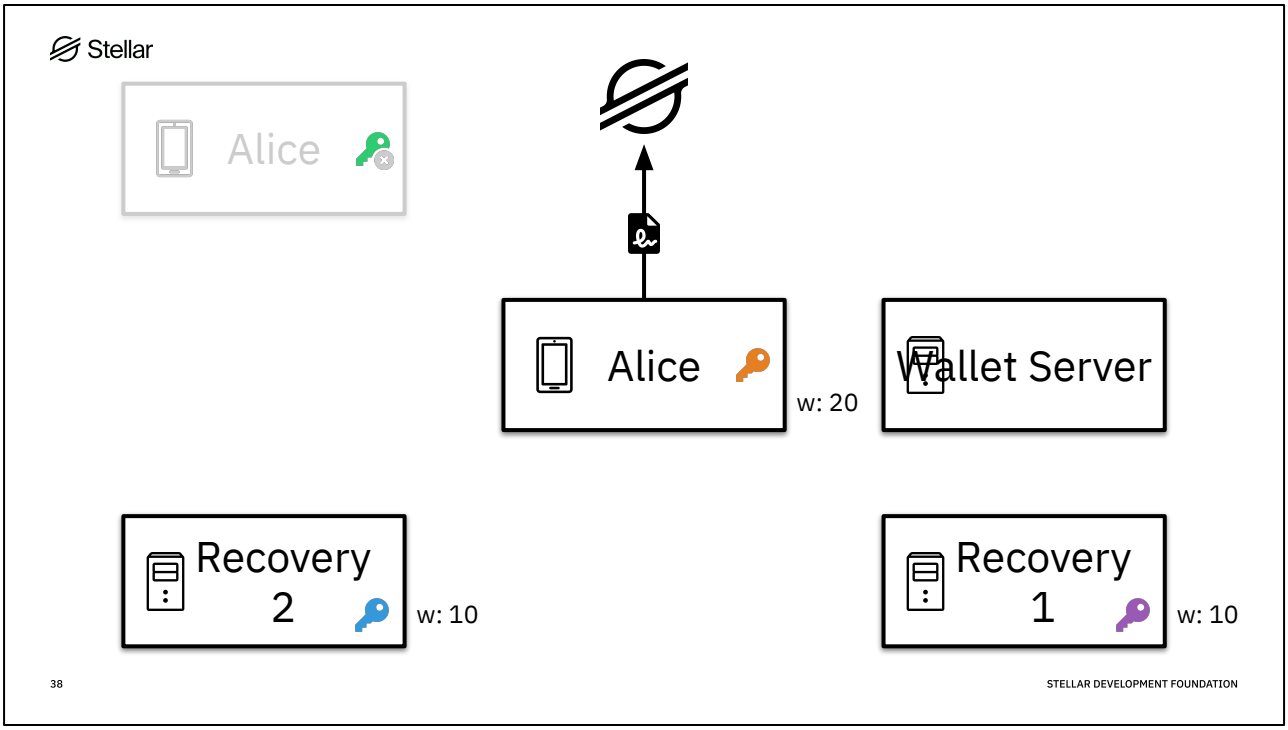
The wallet app talks to the first Recovery server and asks the server to sign a transaction that makes her new device key a signer on the account.



Alice then continues recovery with the Third Party Recovery server.

The third party is operating independently, so Alice authenticates with them independently with an SMS code sent to her phone number or a link or code sent to her email address.

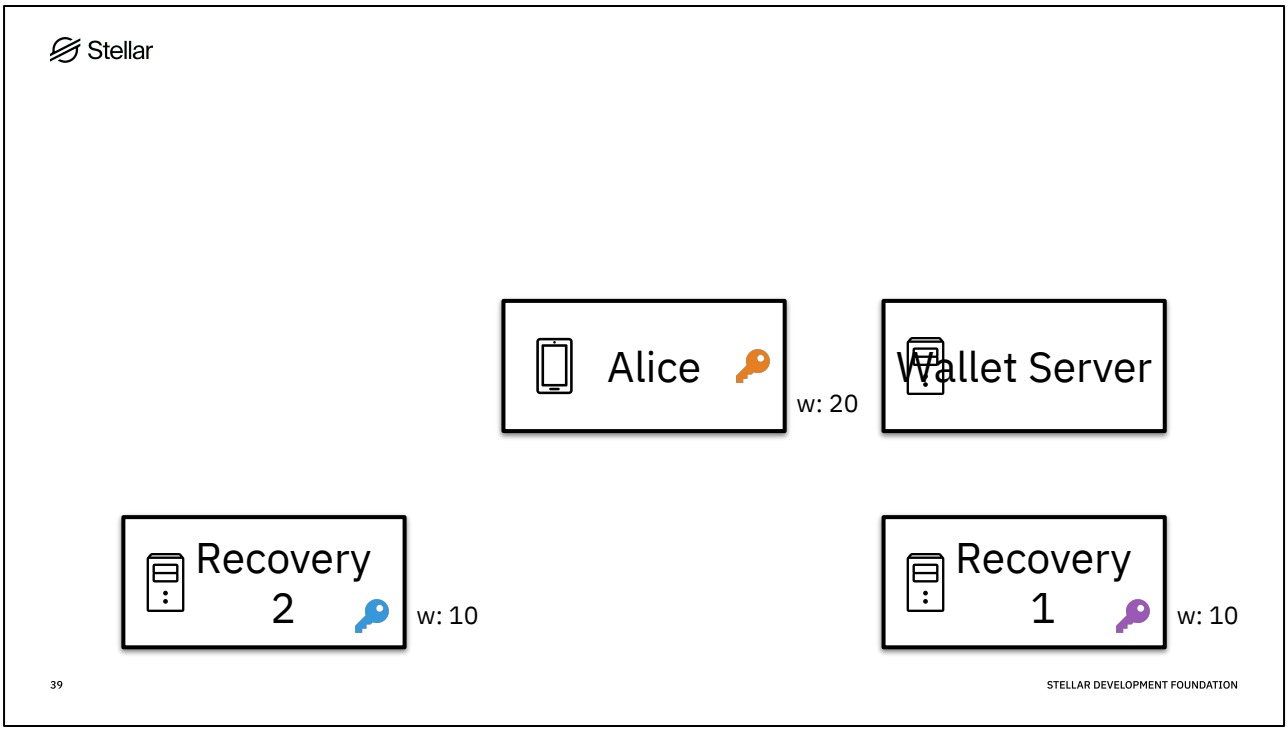
Once Alice has authenticated, the server signs the transaction and returns the signature to the wallet.



The transaction has been authorized with a weight of 20 and can be submitted to the network.

The transaction removes the old signing key that was lost with Alice's previous phone. And adds the new signing key that lives on Alice's new phone.

Alice is now back in control of her Stellar account.



Finally what's important to highlight is that while Alice is in full control of her account, this process can be built into the wallet app and be behind the scenes for her.

Stellar

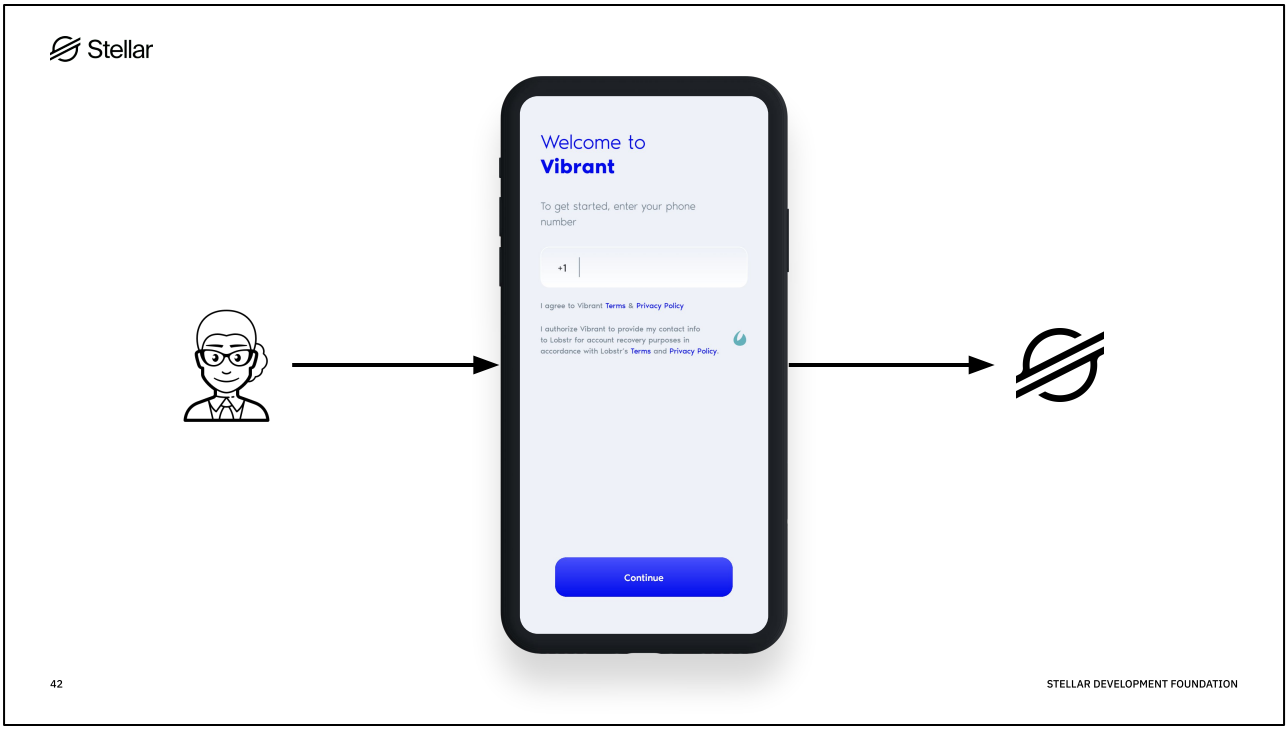
What Alice experiences is the same user experience that we've grown accustomed to in consumer applications.

TRANSITION: Which leads us to...why SEP-30 is valuable right now.

Stellar

Why SEP-30?

Which leads us back to...why SEP-30 is valuable right now.



Great user experience drives adoption, and supporting account recovery through phone and email brings a non-custodial wallet's user experience closer to that of other consumer applications.

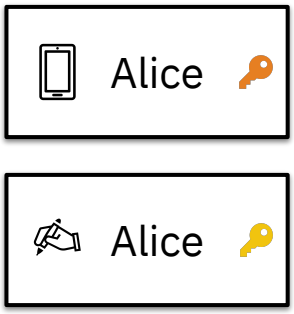
The registration user experience for Alice can be that she signs up like many consumer products, logging in with her phone or email.

And the recovery user experience for Alice is similar requiring only a second SMS or email with a third party recovery provider.

These are patterns that are familiar to users of consumer applications.

TRANSITION: The approach is also flexible...

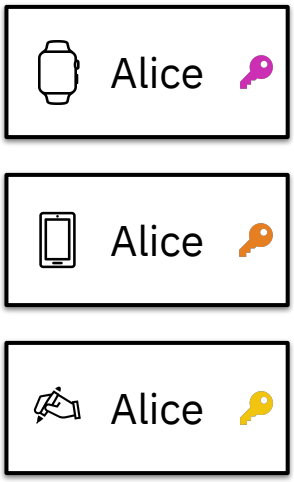
Stellar



TRANSITION: The approach is also flexible…

Allowing the wallet to offer recovery phrases as well as an additional key.

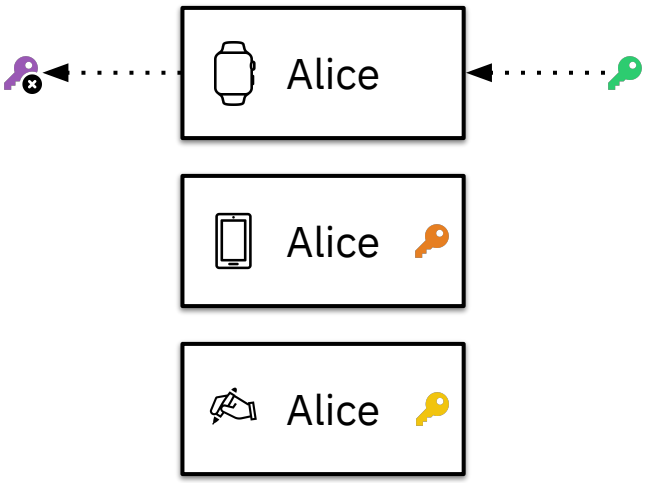
Stellar



Or to use multiple devices where each device generates its own key that never leaves that device.

TRANSITION: This approach also makes dealing with lost devices...simpler.

Stellar



45

STELLAR DEVELOPMENT FOUNDATION

TRANSITION: The approach also makes dealing with lost devices...simpler.

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

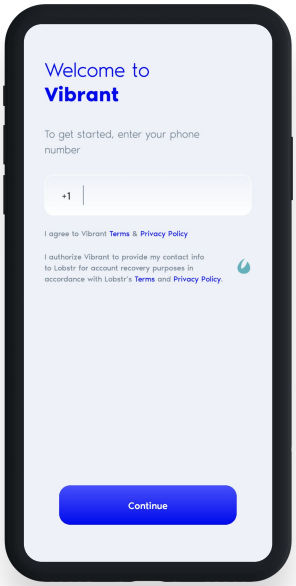
Without interrupting the use of other devices.

Without needing to get a new Stellar account.

And when Alice gets around to replacing her watch it can be given a new key again without interrupting any of her existing devices.

TRANSITION: But the core reason for why SEP-30 really is valuable is...user experience.

Stellar



TRANSITION: But the core reason for why SEP-30 really is valuable is...user experience.

User experience that will drive adoption.

Stellar

Learn More

github.com/stellar/stellar-protocol/blob/master/ecosystem/sep-0030.md

groups.google.com/d/topic/stellar-dev/SFr2dHBZlsY

Thanks all for listening to me talk about SEP-30.

If you'd like to read the proposal itself it's about a 15 minute read.

And there is a thread on the stellar-dev mailing list where you can ask questions, provide feedback, and make suggestions.

Contributions and discussion are welcome.