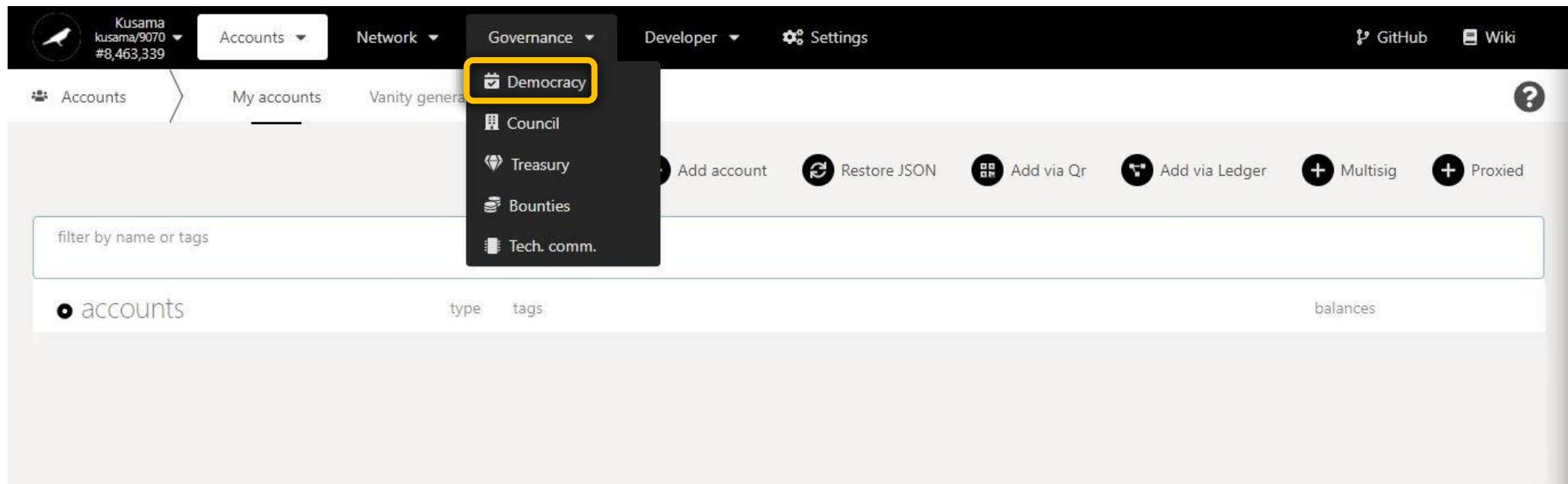


PART IV: Governance

1. Democracy: Explore referendum-related activities.



a) Vote for referenda.

Basic information on technical words used in the DEMOCRACY section.

proposals total referenda total launch period
0 48 1 126 7 days 96%
6 hrs 26 mins

Key information on democracy: **number of proposals**, **number of referenda**, and **time left before a new referendum can be launched**.

referenda
125 preimage 0x4f19b4...304d8278
proposals
No active proposals
external
No external proposal

Referendum pre-image hash.
Countdown to the end of the voting period.
Referendum status (PASS or FAIL)
Referendum turnout

remaining votes
3 days 10 hrs 1,825,3245 KSM
Aye (7) Nay 0.0000 KSM
0.0000 KSM
Referendum-related external links.
+ Image ✓ Vote * 1. Click on the Polkassembly icon to view the discussion thread.

The screenshot shows the Polkassembly interface. At the top, there are navigation links: 'POLKASSEMBLY' (with a logo), 'Discussions', 'On-chain ▾', 'Kusama ▾', and 'anaelleltd ▾'. A blue header bar contains the text '2. Read the information on the referendum.' Below this, a proposal card for '#125 Upgrade Statemine runtime to v2' is displayed. The card indicates the proposal is 'Started'. The description of the proposal states: 'The proposal aims to upgrade Statemine runtime to v2: dispatching an XCM call from the relay chain governance to authorise parachainSystem.authorizeUpgrade(hash). Among other changes, the proposal will enable permissionless asset creation. After approval, the proposal will be voted in referenda queue by the community. Once on-chain governance passes this, anyone can submit an unsigned transaction to submit the actual code and upgrade the runtime (it will have to match the previously agreed hashes).'. A bulleted list provides links for XCM call hash, parachainSystem.authorizeUpgrade(hash) hash, and the runtime file. Below the list, a message encourages voting. At the bottom of the card are 'like' and 'dislike' buttons, a 'Subscribe' button, and a 'Track' button. To the right of the proposal card is a green box titled 'Passing' with a checkmark icon. It shows the results: 'Aye: 1 825.52' (green bar), 'Nay: 0.00', 'Failing threshold: 1 825.52', 'Turnout (0.01%) 1 825.43 KSM', 'Aye 1 825.43 KSM', and 'Nay 0.00 KSM'. A large green arrow points to a pink 'Vote' button in the green box. A blue box at the bottom right contains the text: '3. If you are registered on Polkassembly, you can click on vote to complete the procedure. Otherwise, go back to Polkadot-JS Apps.'

GUIDE TO POLKADOT-JS – PART IV: Governance

Version 2.0

The screenshot shows the Polkadot.js Governance interface for the Kusama network. The top navigation bar includes links for Accounts, Network, Governance (selected), Developer, Settings, GitHub, and Wiki. Below the navigation is a header with a bird icon, the network name 'kusama/9070 #8,463,339', and tabs for Democracy, Overview (selected), and Dispatch.

Key statistics displayed include:

- proposals: 0 total 48
- referenda: 1 total 126
- launch period: 7 days 6 hrs 26 mins (96% complete)

Two main sections are shown:

- referenda**: A list of 125 referenda, with one entry highlighted:
 - preimage: 0x4f19b4...304d8278
 - remaining: 3 days 10 hrs 49,627 blocks
 - votes: Aye (7) 1,825,3245 KSM, Nay 0,0000 KSM
 - Buttons: + Image, ✓ Vote, and other icons
- proposals**: No active proposals listed.
- external**: No external proposals listed.

A green arrow points to the 'Vote' button in the referendum details, with the text '4. Click on vote.' overlaid in a blue box.

The screenshot shows the Polkadot.js Democracy interface. On the left, there's a sidebar with a Kusama icon, the text "kusama/9070 #8,463,363", and a "Democracy" section. Below that is a "proposals" section with a count of 0. In the center, a modal window titled "vote on proposal" is open for proposal #125. The proposal details say "No execution details available for this proposal". The voter information shows "vote with account ANAELLE LTD@KSM". The voting form has a "vote value" input set to "0.1" and a dropdown menu currently set to "KSM". Below the input fields, it says "conviction 0.1x voting balance, no lockup period". To the right of the modal, there's a blue box with the instruction "5. Follow on-screen instructions carefully." and a green box with detailed information about conviction locks. At the bottom of the modal, there are buttons for "Cancel", "Vote Nay", and "Vote Aye".

6. Enter a KSM amount for your vote.

7. Click on the dropdown arrow to view more conviction setups.

5. Follow on-screen instructions carefully.

If this proposal is passed, the changes will be applied via dispatch and the deposit returned.

The vote will be recorded for this account. If another account delegated to this one, the delegated votes will also be counted.

The balance associated with the vote will be locked as per the conviction specified and will not be available for transfer during this period.

Conviction locks do overlap and is additive, meaning that funds locked during a previous vote can be locked again.

vote on proposal

#125 No execution details available for this proposal

vote with account ANAELLE LTD@KSM

vote value 0.1

voting balance 1.7535 KSM

KSM

conviction 0.1x voting balance, no lockup period

Cancel

Vote Nay

Vote Aye

96%

Submit proposal

GUIDE TO POLKADOT-JS – PART IV: Governance

Version 2.0

The screenshot shows the Polkadot.js Democracy interface. On the left sidebar, there are tabs for Democracy, Reference, and Proposals. The Democracy tab is selected. In the main area, a proposal titled "#125 No execution details available for this proposal" is displayed. A dropdown menu titled "vote with account" shows "ANAEILLE LTD@KSM" as the selected account. Below this, the "vote value" is set to "0.1". To the right, it says "voting balance 1.7535 KSM" and "KSM". A dropdown menu for conviction selection lists the following options:

- 0.1x voting balance, no lockup period (selected)
- 1x voting balance, locked for 1x enactment (8.00 days)
- 2x voting balance, locked for 2x enactment (16.00 days)
- 3x voting balance, locked for 4x enactment (32.00 days)
- 4x voting balance, locked for 8x enactment (64.00 days)
- 5x voting balance, locked for 16x enactment (128.00 days)

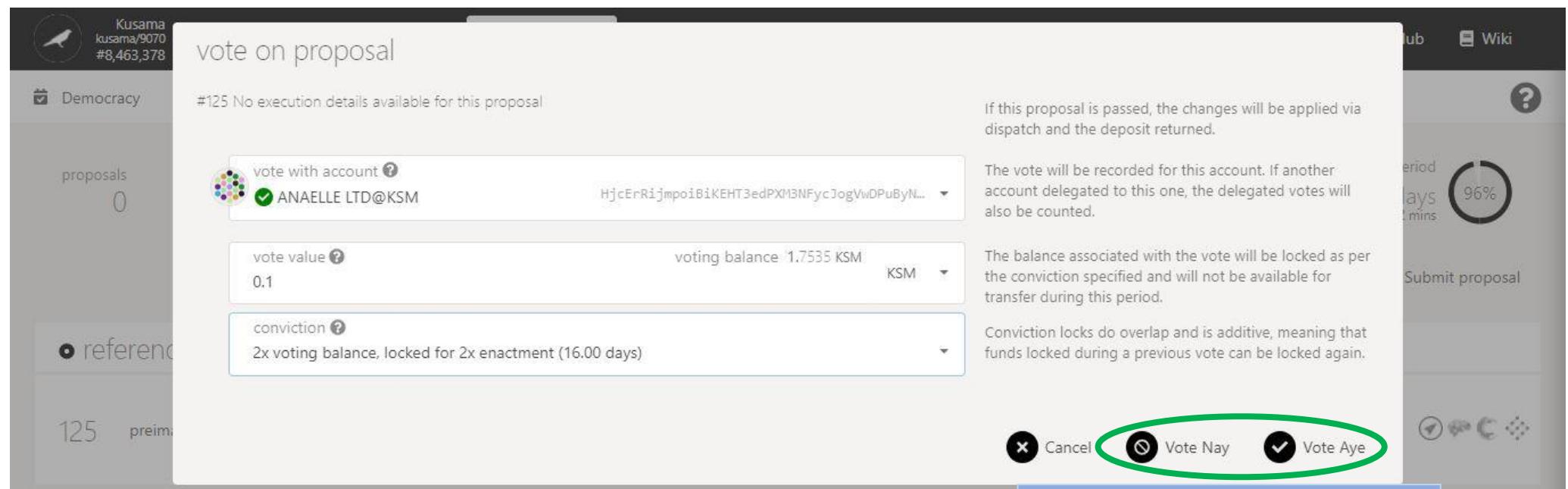
To the right of the conviction dropdown, explanatory text states: "If this proposal is passed, the changes will be applied via dispatch and the deposit returned." Another text block says: "The vote will be recorded for this account. If another account delegated to this one, the delegated votes will also be counted." Below these, it says: "The balance associated with the vote will be locked as per the conviction specified and will not be available for transfer during this period." Further down, it notes: "Conviction locks do overlap and is additive, meaning that funds locked during a previous vote can be locked again." At the bottom right of the interface are buttons for "Cancel", "Vote Nay", and "Vote Aye".

8. Select one conviction setup for your vote.

Note: KSM balances locked through democracy votes must be unlocked at the end of the lockup period to become transferrable.

GUIDE TO POLKADOT-JS – PART IV: Governance

Version 2.0



9. Click on **Vote Nay** or **Vote Aye** to continue the procedure.

Nature of the transaction.

The screenshot shows the Polkadot-JS extension interface for Kusama. On the left, there's a sidebar with 'Democracy' selected, showing 0 proposals. The main area has a title 'authorize transaction'. Below it, a message says 'Sending transaction democracy.vote(ref_index, vote)'. A note states: 'Vote in a referendum. If vote.is_aye(), the vote is to enact the proposal; otherwise it is a vote to keep the status quo.' A green box highlights the fee information: 'Fees of 45.9995 micro KSM will be applied to the submission'. To the right, a blue box contains the text '10. Check the transaction fees.' Below this, it says 'sending from my account ANAELLE LTD@KSM'. There's a toggle switch for 'Do not include a tip for the block author'. Underneath is a call hash: '0x91fe984fc027ffd24f27555a5e6cb1cae0a3e230804e380346a2ead15e100ed9'. At the bottom, there are 'Sign and Submit' and 'Cancel' buttons. A green arrow points from the text 'Nature of the transaction.' to the 'democracy.vote queued' status bar at the top right. Another green circle highlights the 'Sign and Submit' button.

11. Click on **Sign & submit** to continue the procedure.

The screenshot shows the Polkadot-JS extension interface for the Kusama network. A green arrow points from the text "Summary of the transaction sent via the Polkadot-JS extension." to the transaction details in the center. Another green arrow points from the text "13. Click on Sign the transaction to complete the procedure." to the "Sign the transaction" button at the bottom of the sign-in dialog. A third green arrow points from the text "12. Enter your account's password and tick the box to remember your password, if necessary." to the password input field and the "Remember my password for the next 15 minutes" checkbox. A yellow box labeled "Progress of the transaction." contains a progress bar showing 96% completion with 7 days and 6 hrs 20 mins remaining. The transaction details include the sender (Anaelle LTD @KSM), chain (Kusama), version (9070), nonce (103), method (democracy.vote(ref_index, vote)), and lifetime (mortal, valid from 8,463,386 to 8,463,450).

Summary of the transaction sent via the Polkadot-JS extension.

13. Click on Sign the transaction to complete the procedure.

12. Enter your account's password and tick the box to remember your password, if necessary.

Progress of the transaction.

7 days 6 hrs 20 mins 96%

Transaction

Anaelle LTD @KSM HjcErRijmpoiBiKEHT3edPXM3NFycJogVwDPuByNe7hv9Ae

from https://cloudflare-ipfs.com/ipns/dotapps.io/#/democracy
chain Kusama
version 9070
nonce 103
method ► democracy.vote(ref_index, vote)
info ► Vote in a referendum. If `vote.is_aye()`, the vote is to enact the ...
lifetime mortal, valid from 8,463,386 to 8,463,450

PASSWORD FOR THIS ACCOUNT
Sign the transaction

GUIDE TO POLKADOT-JS – PART IV: Governance

Version 2.0

The screenshot shows the Polkadot.js Governance interface. At the top, there are navigation tabs: Democracy, Overview (which is selected), and Dispatch. The top bar also includes account information (kusama/9070, #8,463,441), network selection (Network), governance (Governance), developer (Developer), settings (Settings), GitHub link, and a Wiki link.

Below the tabs, there are two summary sections: "proposals" (0 total 48) and "referenda" (1 total 126). To the right, a progress bar indicates a "launch period" of 7 days (6 hrs 15 mins) with a completion of 96%.

The main content area displays a referendum with the following details:

- referenda**: 125 preimage 0x4f19b4...304d8278
- remaining**: 3 days 10 hrs 49,525 blocks
- votes**:
 - Aye (8) 1,825.5245 KSM
 - Nay 0.0000 KSM

On the right side of the referendum card, there are buttons for "Submit preimage" and "Submit proposal". Below the referendum card, there is a section for "proposals" which currently says "No active proposals".

Two callouts are present:

- 14. Your vote has been included!** (highlighting the Aye vote count)
- 15. Click on the dropdown arrow to view your voting details.** (highlighting the dropdown arrow next to the Aye vote count)

GUIDE TO POLKADOT-JS – PART IV: Governance

Version 2.0

The screenshot shows the Polkadot.js Governance interface for the Kusama network. The top navigation bar includes links for Accounts, Network, Governance (selected), Developer, Settings, GitHub, and Wiki. Below the navigation is a header with tabs for Democracy, Overview (selected), and Dispatch. The Overview tab displays statistics: 0 proposals (total 48), 1 referenda (total 126), and a launch period of 7 days, 6 hrs 15 mins, which is 96% complete.

The main content area shows a list of referenda. One referendum is listed under the "referenda" tab, with a status of "Aye (8)" and a total amount of 1,825,5245 KSM. The list of voters includes:

- LIPENG (1x - 1,800,000 KSM)
- RTTI-5220 (1x - 20,000 KSM)
- PLAYING WITH DL (1x - 4,000 KSM)
- H2CFDT...T2JHXY (1x - 1,3233 KSM)
- ANAEILLE LTD@KS (2x - 0,1000 KSM) - This account is highlighted with a green arrow pointing to it from a callout box.
- DiUCbg...nVbvsj (0.1x - 0,0116 KSM)
- Cp49ff...3b5PRT (0.1x - 0,0012 KSM)
- Dqxa5p...wgXHAp (0.1x - 0,0000 KSM)

Callout box text: 16. Your account address, lockup period and voting amount are displayed!

b) Clear expired democracy locks.

1. Click **Accounts**.



Kusama
kusama/9080
#8,754,594

Accounts Network Governance Developer Settings GitHub Wiki

Accounts My accounts Vanity generator ?

Add account Restore JSON Add via Qr Add via Ledger Multisig Proxied

filter by name or tags

accounts	type	tags	balances
ALTD@KSM (EXTENSION)	injected	no tags	20.3948 KSM send :
ANAEILLE LTD@KSM	injected	no tags	2,7866 KSM send : 23.1814 KSM

2. Click on the
dropdown arrow to
view your balances.

GUIDE TO POLKADOT-JS – PART IV: Governance

Version 2.0

The screenshot shows the Polkadot-JS extension interface with the following details:

- Top Bar:** Kusama kusama/9080 #8,754,603, Accounts (dropdown), Network (dropdown), Governance (dropdown), Developer (dropdown), Settings.
- Header:** GitHub, Wiki, ?
- Navigation:** Accounts (selected), My accounts, Vanity generator.
- Account Actions:** Add account, Restore JSON, Add via QR, Add via Ledger, Multisig, Proxied.
- Search:** filter by name or tags.
- Table Headers:** accounts, type, tags, balances.
- Accounts:**
 - ALTD@KSM (EXTENSION)**: injected, no tags, 20.3948 KSM, send, more options.
 - ANAELLE LTD@KSM**: checked, injected, no tags, 23.1814 KSM, send, more options.
 - balances: 2.7866 KSM (transferrable), 1.7783 KSM (locked), 0.9000 KSM (reserved), 0.1083 KSM (democracy).
- Callout:** 3. Your democracy balance is now visible! (highlighting the 0.1000 KSM entry for the ANAELLE LTD@KSM account).

GUIDE TO POLKADOT-JS – PART IV: Governance

Version 2.0

The screenshot shows the Polkadot.js extension interface. At the top, there's a navigation bar with tabs for Accounts, Network, Governance, Developer, and Settings. The Accounts tab is selected, showing "My accounts" and a "Vanity generator". Below this, there are buttons for adding accounts, restoring from JSON, adding via QR, adding via Ledger, creating multisig, and creating proxied accounts. A search bar allows filtering by name or tags. The main area displays a list of accounts:

	accounts	type	tags
★	ALTD@KSM (EXTENSION)	injected	no tags
★	ANAEILLE LTD@KSM	injected	no tags

A blue callout box with a green arrow points to the three vertical dots next to the ANAEILLE LTD@KSM account, with the text: "5. Click on the 3 vertical dots to view Account settings." Another blue callout box with a green arrow points to the lock status icon for the ANAEILLE LTD@KSM account, with the text: "4. Hover on the icon to check the lock status." The lock status for the ANAEILLE LTD@KSM account is shown as "#125 0.1000 KSM Locked2x lock expired".

5. Click on the 3 vertical dots to view Account settings.

4. Hover on the icon to check the lock status.

The screenshot shows the Polkadot-JS UI interface. At the top, there is a navigation bar with links for Accounts, Network, Governance, Developer, and Settings. Below the navigation bar, the user's account information is displayed: Kusama, kusama/9080, #8,754,616. The main content area is titled "My accounts" and shows two accounts listed: "ALTD@KSM (EXTENSION)" and "ANAEILLE LTD@KSM". The "ANAEILLE LTD@KSM" account has a green checkmark next to its name. A blue callout box with the text "6. Click on Clear expired democracy locks to remove your votes." is overlaid on the screen. A green arrow points from this callout to the "Clear expired democracy locks" option in a context menu that is open on the right side of the screen. The context menu also includes other options like Set on-chain identity, Set on-chain sub-identities, Make recoverable, Initiate recovery for another, Delegate democracy votes, and Add proxy. At the bottom right of the screen, there is a summary of the account's balance: 23.1814 KSM, with breakdowns for transferrable, locked (0.9000 KSM), reserved (0.1083 KSM), and democracy (0.1000 KSM). There are also "send" and "more" buttons.

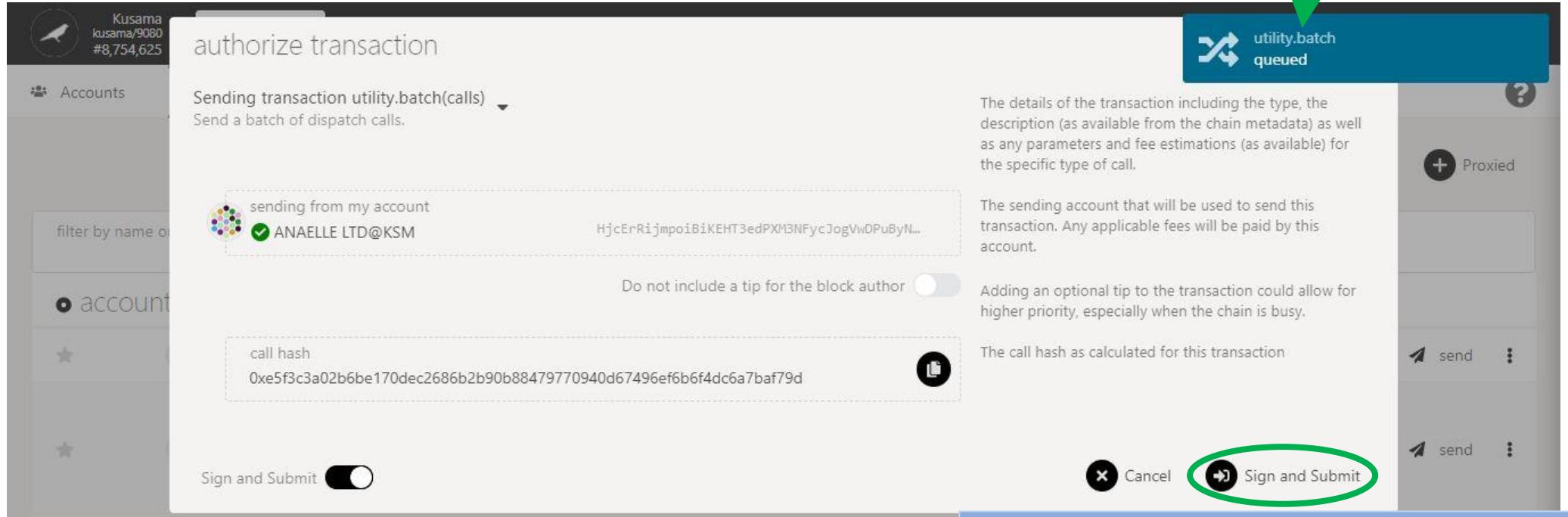
6. Click on **Clear expired democracy locks** to remove your votes.

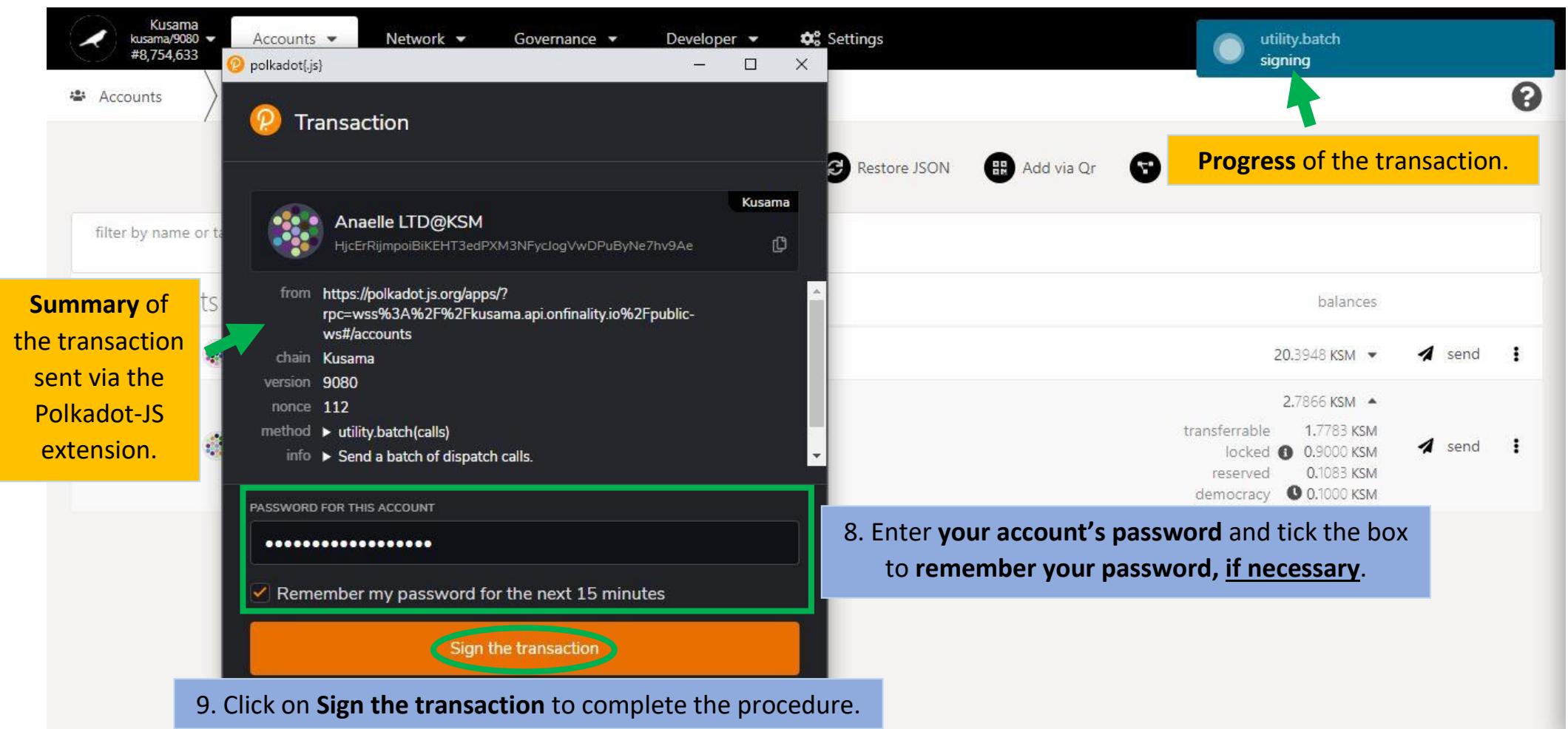
Ledger

- Set on-chain identity
- Set on-chain sub-identities
- Clear expired democracy locks**
- Make recoverable
- Initiate recovery for another
- Delegate democracy votes
- Add proxy

transferrable
locked 0.9000 KSM
reserved 0.1083 KSM
democracy 0.1000 KSM

23.1814 KSM

Nature of the transaction.**7. Click on **Sign & submit** to continue the procedure.**



GUIDE TO POLKADOT-JS – PART IV: Governance

Version 2.0

The screenshot shows the Polkadot-JS Accounts interface. At the top, there's a navigation bar with links for Accounts, Network, Governance, Developer, and Settings. On the far right of the bar are GitHub and Wiki links. Below the navigation bar, there are tabs for Accounts (selected), My accounts, and Vanity generator. A question mark icon is on the far right of this section.

Below the tabs, there are several action buttons: Add account, Restore JSON, Add via QR, Add via Ledger, Multisig, and Proxied. There's also a search bar labeled "filter by name or tags".

The main content area displays two accounts:

accounts	type	tags	balances
★ ALTD@KSM (EXTENSION)	injected	no tags	2.7866 KSM transferrable 1.7782 KSM locked 0.9000 KSM reserved 0.1083 KSM
★ ANAELLE LTD@KSM	injected	no tags	23.1814 KSM

A blue callout box with the text "10. Your democracy locks have been cleared!" is positioned above the second account. The "transferrable" balance for the first account is highlighted with a green border.

c) Submit referendum pre-images.

Kusama
kusama/9080
#8,583,196

Accounts ▾ Network ▾ Governance ▾ Developer ▾ Settings

GitHub Wiki

Democracy Overview Dispatch ?

proposals total 0 48

referenda total 0 128

launch period 7 days 5 days 22 hrs 15%

+ Submit preimage + Submit proposal

1. Click on **Submit preimage**.

referenda
No active referendums

proposals
No active proposals

external
No external proposal

Referendum preimage in detail:
A **referendum preimage** is the first of two steps when proposing a public referendum that the community can vote on.
The preimage hash is required to enact/implement a referendum proposal once it has been passed/approved by voting.

2. Follow on-screen instructions carefully.

This account will pay the fees for the preimage, based on the size thereof.

The image (proposal) will be stored on-chain against the hash of the contents.

When submitting a proposal the hash needs to be known. Proposals can be submitted with hash-only, but upon dispatch the preimage needs to be available.

The calculated storage costs based on the size and the per-bytes fee.

Cancel + Submit preimage

submit preimage

send from account ? **ANAELE LTD@KSM** transferrable 0.1389 KSM

propose ? system setCode(code)

code: Bytes
0x prefixed hex, e.g. 0x1234 or ascii data

file upload

preimage hash ?
0x0e5751c026e543b2e8ab2eb06099daa1d1e5df47778f7787faab45cdf12fe3a8

calculated storage fee ? 0 pico

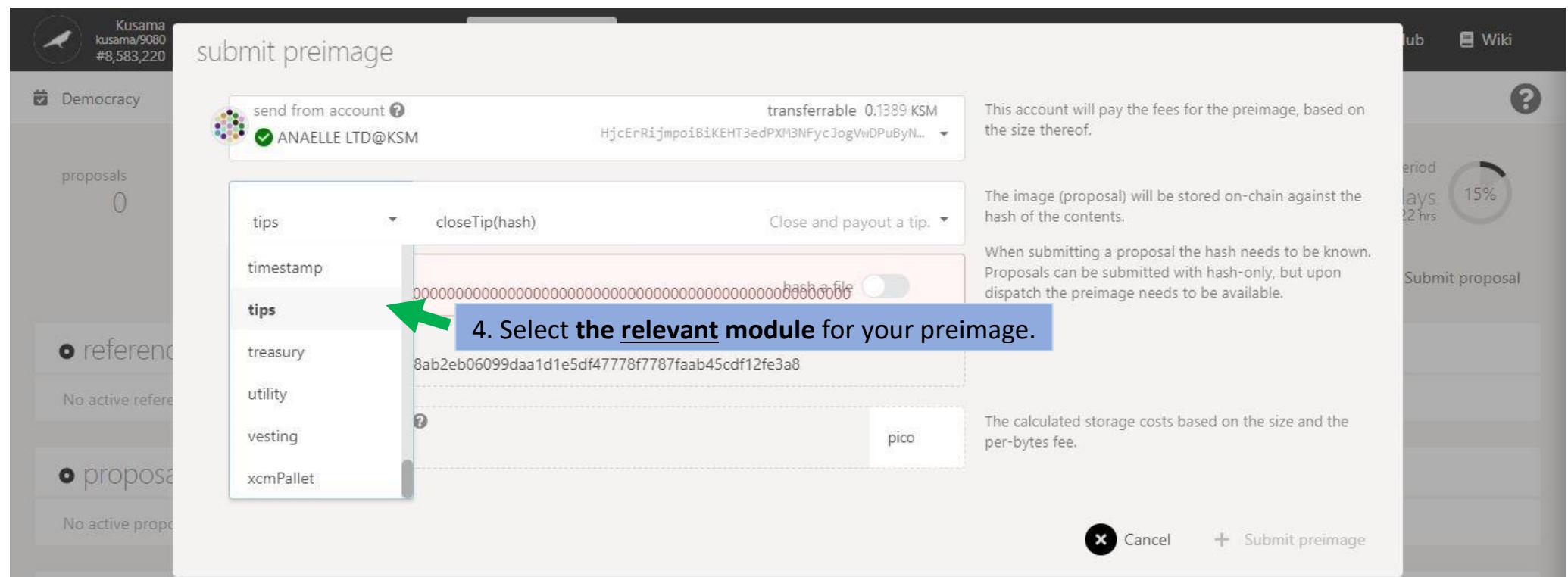
3. Click on the dropdown arrow to view the list of runtime modules.

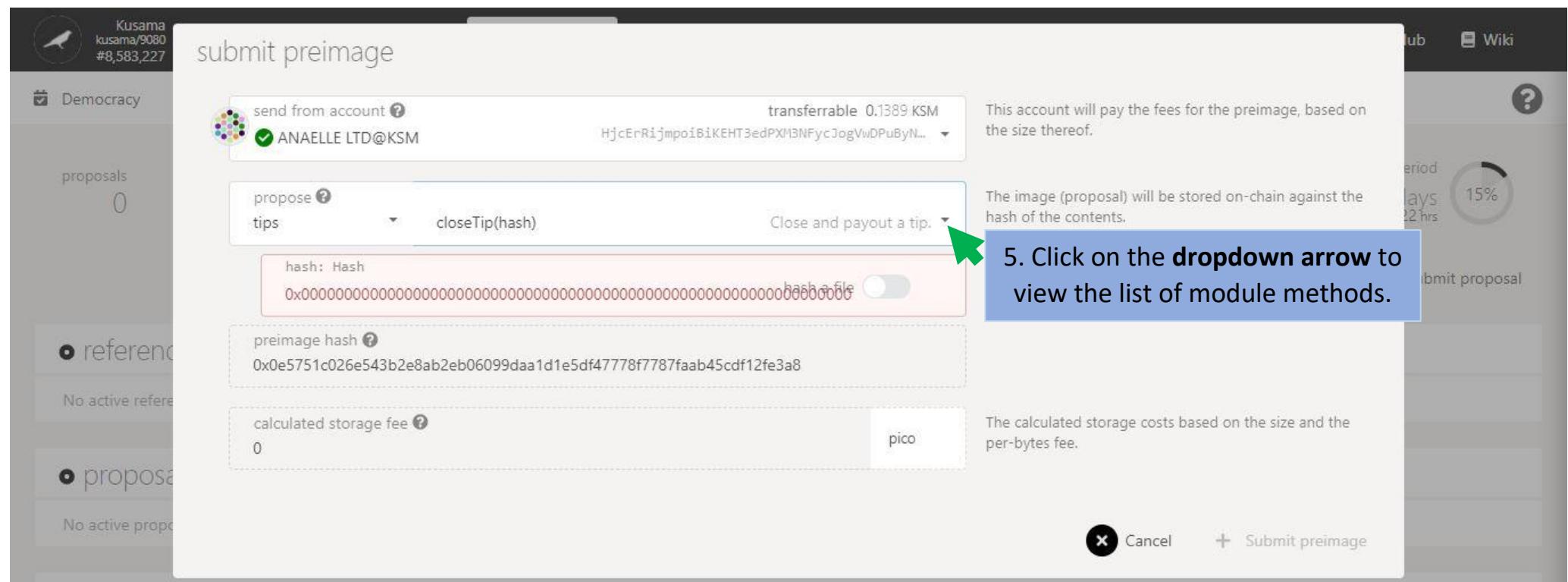
No active references

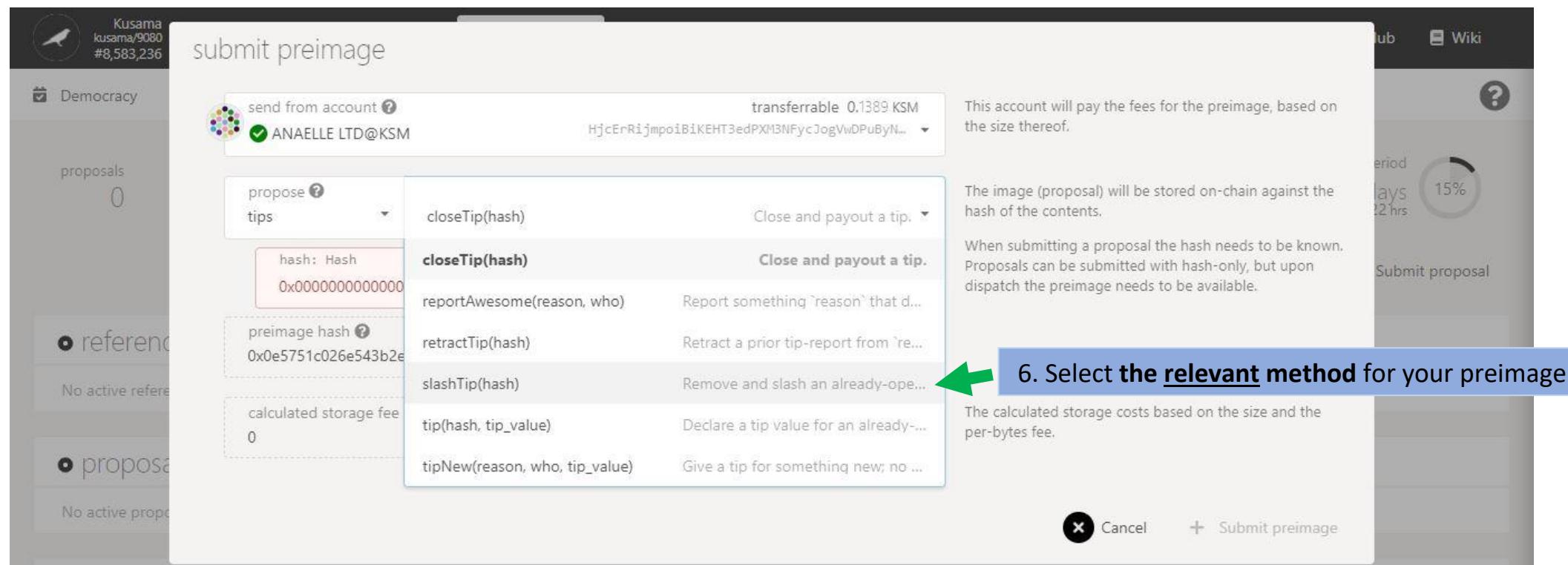
No active proposals

Wiki

period days 15% 2 hrs Submit proposal





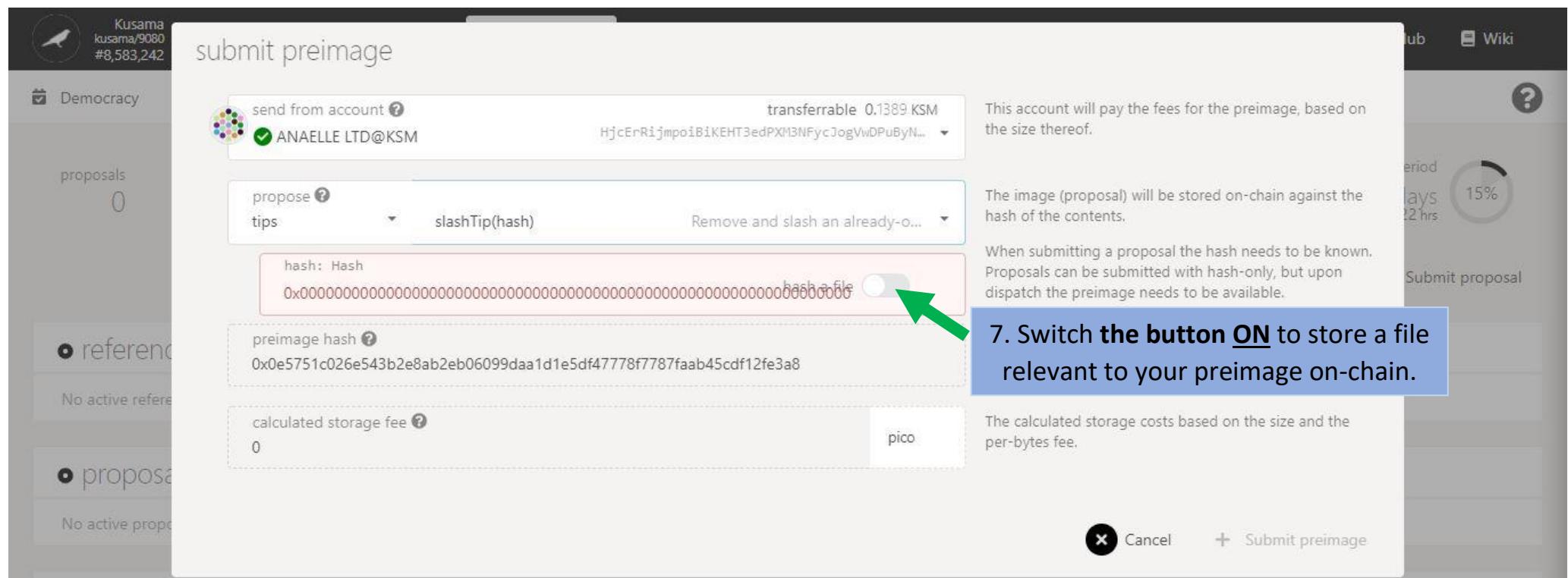


The screenshot shows the Polkadot.js Democracy module with the title "submit preimage". The top bar displays account information: "kusama/kusama/9080 #8,583,236". The left sidebar shows "Democracy" selected, with "proposals 0" and "referendum 0". The main area has a heading "send from account" with "ANAEILLE LTD@KSM" selected. Below it, a dropdown menu titled "tips" lists several methods:

- closeTip(hash)** (highlighted with a green arrow): Close and payout a tip.
- reportAwesome(reason, who)
- retractTip(hash)
- slashTip(hash)
- tip(hash, tip_value)
- tipNew(reason, who, tip_value)

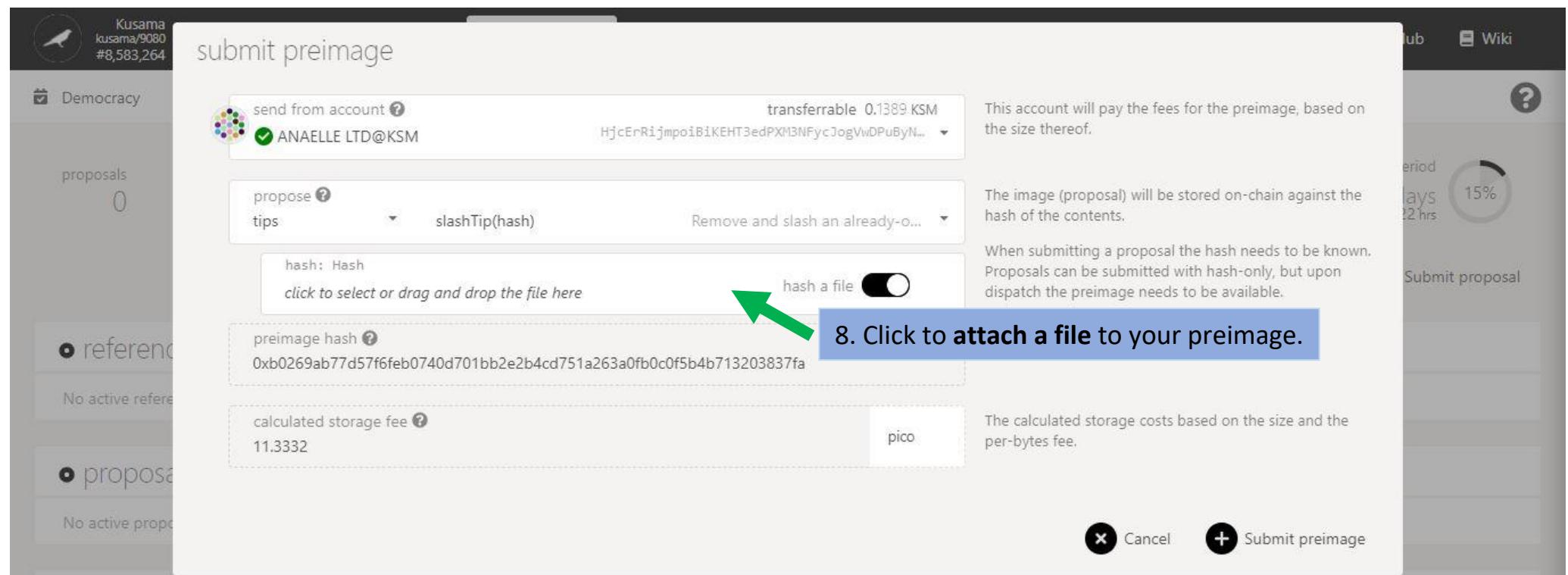
Details for the selected method ("closeTip(hash)") are shown: "transferrable 0.1389 KSM" and the hex hash "HjcErRijmpoiBiKEHT3edPXM3NFycJogVwDPuByN...". To the right, explanatory text states: "This account will pay the fees for the preimage, based on the size thereof." and "The image (proposal) will be stored on-chain against the hash of the contents." Below the dropdown, a note says: "When submitting a proposal the hash needs to be known. Proposals can be submitted with hash-only, but upon dispatch the preimage needs to be available." At the bottom right are "Cancel" and "Submit preimage" buttons.

6. Select the relevant method for your preimage.



GUIDE TO POLKADOT-JS – PART IV: Governance

Version 2.0



The screenshot shows the Polkadot.js wallet interface with the title "submit preimage". On the left sidebar, there are sections for Democracy, proposals (0), references, and proposals (No active proposals). The main area has a heading "send from account" with "ANAELE LTD@KSM" selected. Below it, a dropdown menu shows "propose tips slashTip(hash)". A green box highlights the "hash: Hash" input field containing the value "0x8fa6f9ef7707faf83fa13e05a4fea81e79031166acaaff69ba5bd8962cb7946e". To the right, a note says "This account will pay the fees for the preimage, based on the size thereof." Further down, another note says "The image (proposal) will be stored on-chain against the hash of the contents." and "When submitting a proposal the hash needs to be known. Proposals can be submitted with hash-only, but upon dispatch the preimage needs to be available." At the bottom, a blue box contains the text "9. Your file has been hashed and is ready to be stored on-chain!". Below this, another blue box contains the text "10. Check the storage fees." with a green arrow pointing to the "calculated storage fee" field which shows "11.3332 pico". At the bottom right are "Cancel" and "Submit preimage" buttons.

submit preimage

send from account ?
ANAELE LTD@KSM

transferrable 0.1389 KSM
HjcErRijmpoiBiKEHT3edPXM3NFycJogVwDPuByN...

This account will pay the fees for the preimage, based on the size thereof.

propose ?
tips slashTip(hash)

Remove and slash an already-o...
hash: Hash
0x8fa6f9ef7707faf83fa13e05a4fea81e79031166acaaff69ba5bd8962cb7946e

The image (proposal) will be stored on-chain against the hash of the contents.

When submitting a proposal the hash needs to be known. Proposals can be submitted with hash-only, but upon dispatch the preimage needs to be available.

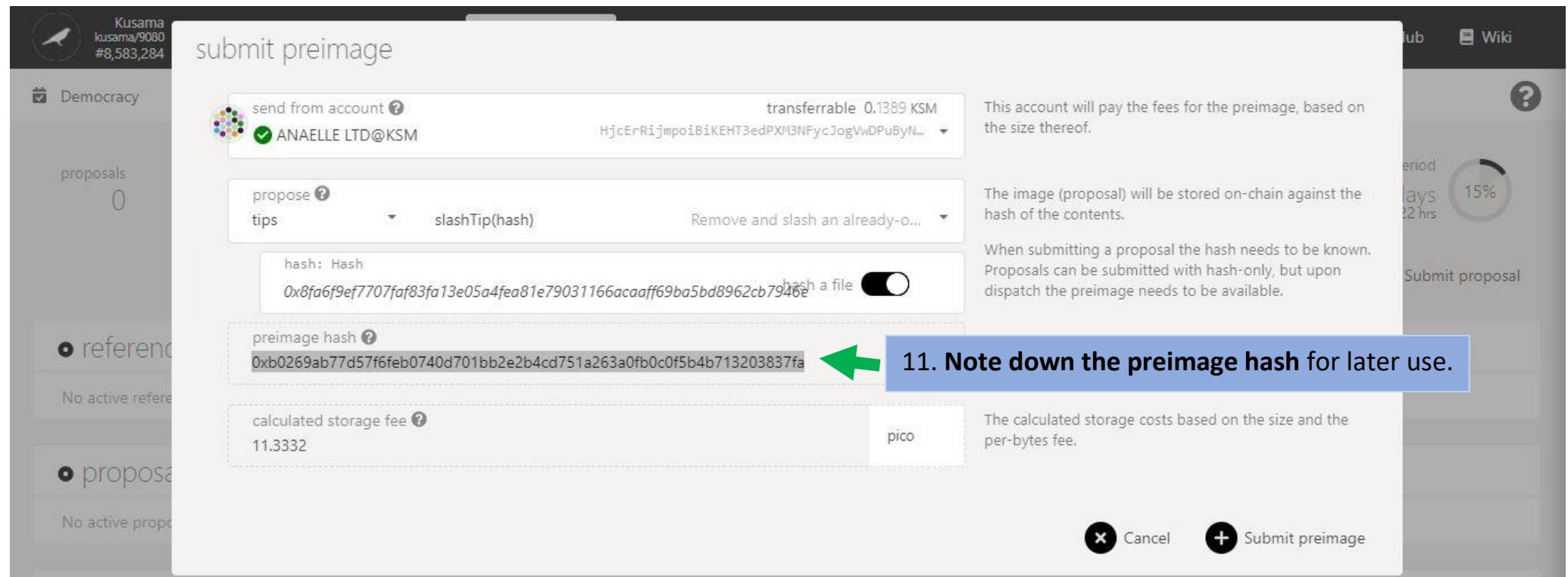
9. Your file has been hashed and is ready to be stored on-chain!

calculated storage fee ?
11.3332 pico

The calculated storage costs based on the size and the per-bytes fee.

Cancel Submit preimage

10. Check the storage fees.



Nature of the transaction.

The screenshot shows the Polkadot.js extension interface for the Kusama network. The main title is "authorize transaction". Below it, the transaction details are listed: "Sending transaction democracy.notePreimage(encoded_proposal)". A note says "Register the preimage for an upcoming proposal. This doesn't require the proposal to be in the dispatch queue but does require a deposit, returned once enacted." A green box highlights the fee information: "Fees of 50.9995 micro KSM will be applied to the submission". To the right, a blue box labeled "12. Check the transaction fees." contains the transaction details: "democracy.notePreimage queued". A green arrow points from the "Nature of the transaction." header to this box. Below the transaction details, there are two options: "Do not include a tip for the block author" (unchecked) and "Adding an optional tip to the transaction could allow for higher priority, especially when the chain is busy." Further down, the "call hash" is shown as "0x10a3f34a56ed47dc111067f6d9bd72c3fc4cdc68ce742473e5c20c42afbb8c87". At the bottom, there are "Sign and Submit" and "Cancel" buttons. The "Sign and Submit" button is circled in green. A blue box labeled "13. Click on Sign & submit to continue the procedure." is positioned at the bottom right.

authorize transaction

Sending transaction `democracy.notePreimage(encoded_proposal)`

Register the preimage for an upcoming proposal. This doesn't require the proposal to be in the dispatch queue but does require a deposit, returned once enacted.

Fees of 50.9995 micro KSM will be applied to the submission

sending from my account
ANAEILLE LTD@KSM

HjcErRijmpoiBiKEHT3edPXMI3NFycJogVwDPuByN...

Do not include a tip for the block author

Call hash
0x10a3f34a56ed47dc111067f6d9bd72c3fc4cdc68ce742473e5c20c42afbb8c87

Sign and Submit

Cancel

Sign and Submit

democracy.notePreimage queued

12. Check the transaction fees.

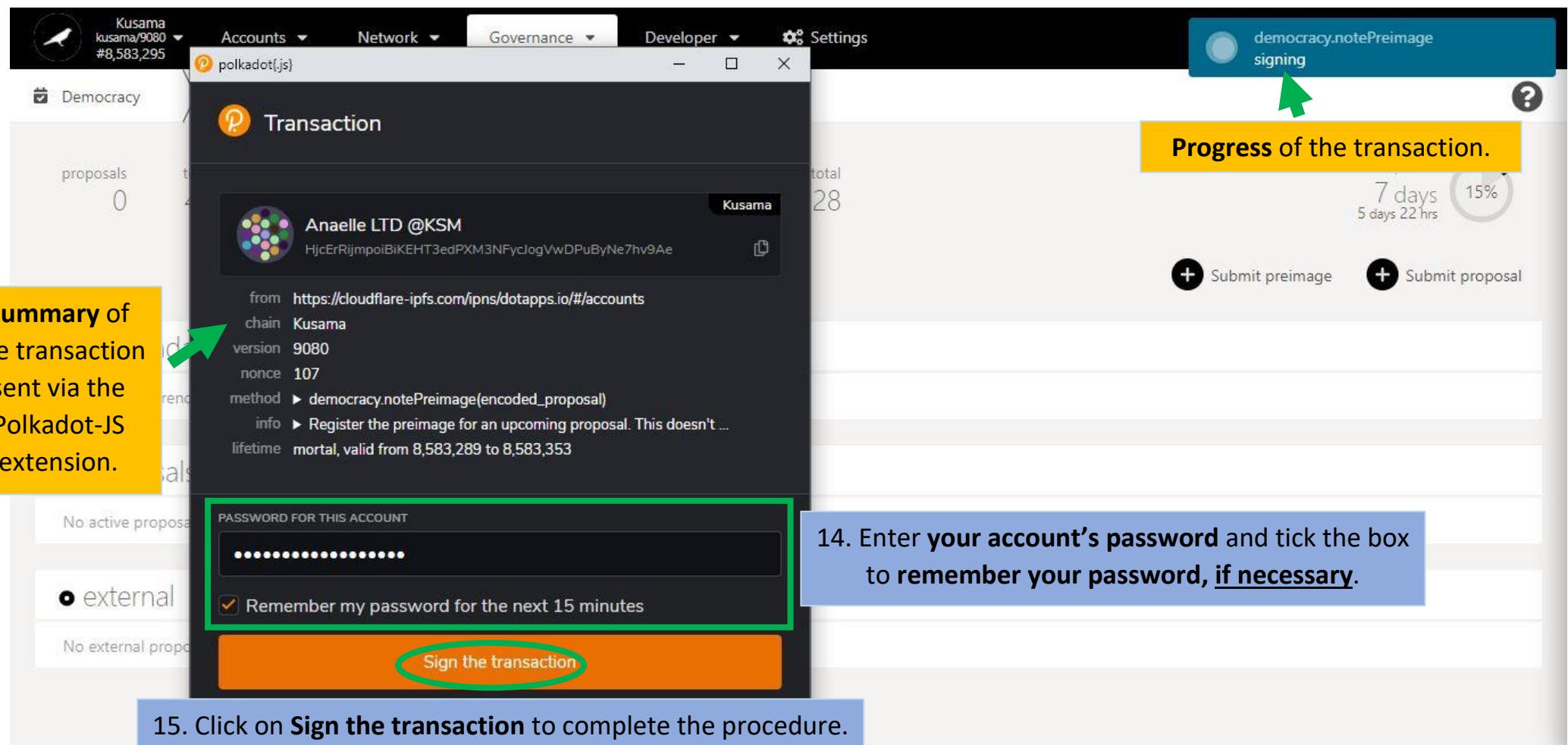
The details of the transaction including the type, the description (as available from the chain metadata) as well as any parameters and fee estimations (as available) for the call.

The sending account that will be used to send this transaction. Any applicable fees will be paid by this account.

Adding an optional tip to the transaction could allow for higher priority, especially when the chain is busy.

The call hash as calculated for this transaction

13. Click on **Sign & submit** to continue the procedure.



d) Submit referendum proposals.

Kusama
kusama/9080
#8,583,300

Accounts ▾ Network ▾ Governance ▾ Developer ▾ Settings

GitHub Wiki

Democracy Overview Dispatch ?

proposals total 0 48

referenda total 0 128

launch period 7 days
5 days 22 hrs 15%

+ Submit preimage + Submit proposal

1. Click on **Submit proposal**.

● referenda
No active referendums

● proposals
No active proposals

● external
No external proposal

Referendum proposal in detail:
A **referendum proposal** is the second of two steps when proposing a public referendum that the community can vote on.
A minimum deposit is required to submit a proposal, but it will be released when the proposal is timetabled as a public referendum.

2. Follow on-screen instructions carefully.

The proposal will be registered from this account and the balance lock will be applied here.

3. Enter the required information.

send from account ? **ANAEILLE LTD@KSM** transferrable 0.1389 KSM
HjcErRijmpoiBiKEHT3edPXN3NFycJogVwDPuByN...

preimage hash ?
0xb0269ab77d57f6feb0740d701bb2e2b4cd751a263a0fb0c0f5b4b713203837fa

locked balance ? **0.01** KSM milli

minimum deposit ? **3.3333**

4. Check that you can afford the minimum deposit.

5. Click on Submit proposal to continue the procedure.

Cancel **Submit proposal**

Nature of the transaction.

The screenshot shows the Polkadot-JS interface for governance. On the left, there's a sidebar with sections for Democracy, proposals (0), references (0), and proposals (0). The main area is titled "authorize transaction" and shows a proposal to "democracy.propose(proposal_hash, value)". It includes a note about fees and the account from which it will be sent. A green box highlights the fee information: "Fees of 51.9995 micro KSM will be applied to the submission". To the right, a blue box contains the instruction "6. Check the transaction fees." Below this, there's a section for a tip to the block author and the call hash (0x37a21d25e966b5102383119a8fad900a9e51c2a5eb5d21f7235326e2d3fd2ca7). At the bottom right are "Cancel" and "Sign and Submit" buttons, with "Sign and Submit" circled in green. A yellow box at the top right says "Nature of the transaction." with a green arrow pointing to the "democracy.propose queued" status in the top right corner of the main window.

authorize transaction

Sending transaction `democracy.propose(proposal_hash, value)`
Propose a sensitive action to be taken.

Fees of 51.9995 micro KSM will be applied to the submission

sending from my account
ANAEILLE LTD@KSM

HjcErRijmpoiBIKEHT3edPXH3NFycJogVwDPuByN...

Do not include a tip for the block author

call hash
0x37a21d25e966b5102383119a8fad900a9e51c2a5eb5d21f7235326e2d3fd2ca7

Sign and Submit

6. Check the transaction fees.

The details of the transaction including the type, the description (as available from the chain metadata) as well as meters and fee estimations (as available) for this type of call.

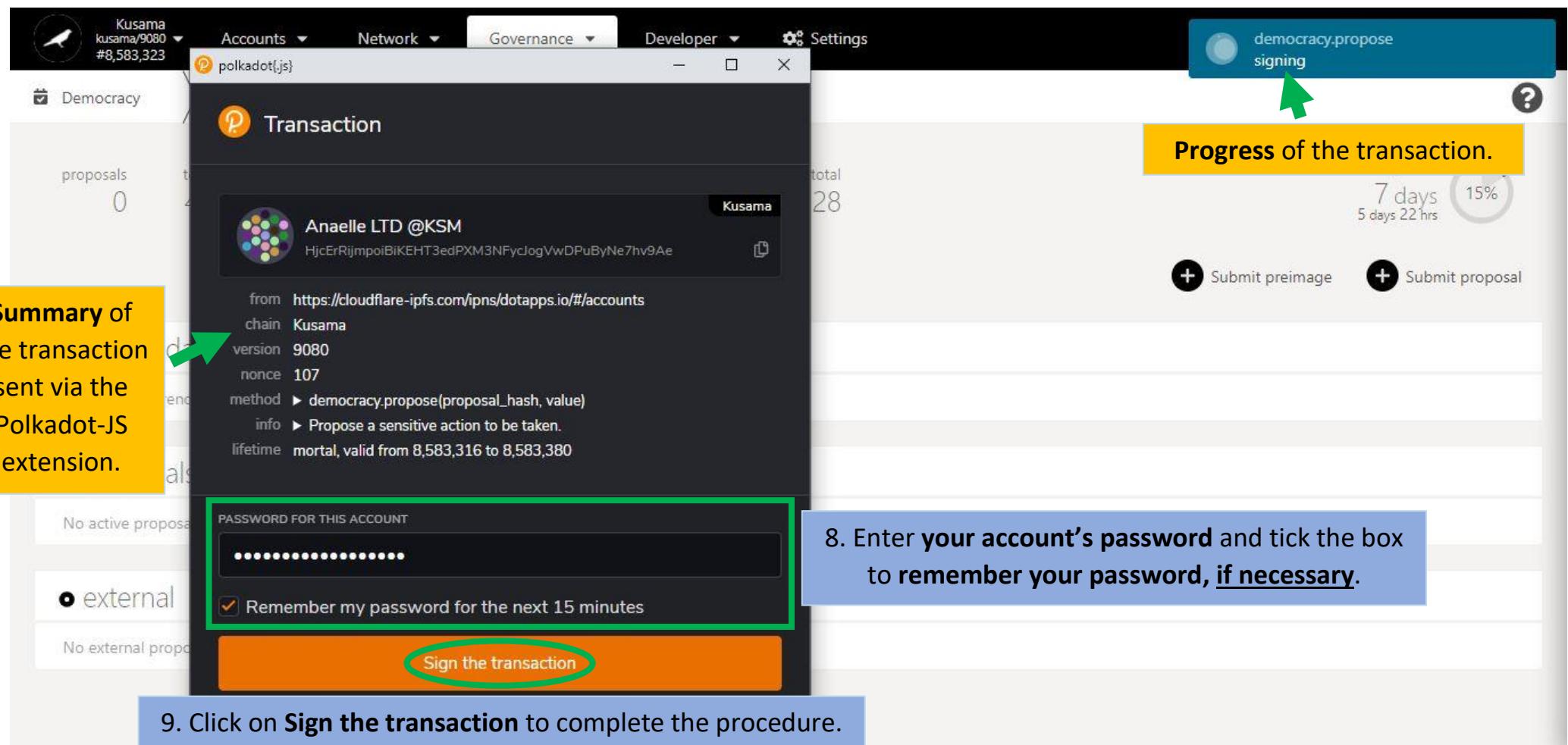
The sending account that will be used to send this transaction. Any applicable fees will be paid by this account.

Adding an optional tip to the transaction could allow for higher priority, especially when the chain is busy.

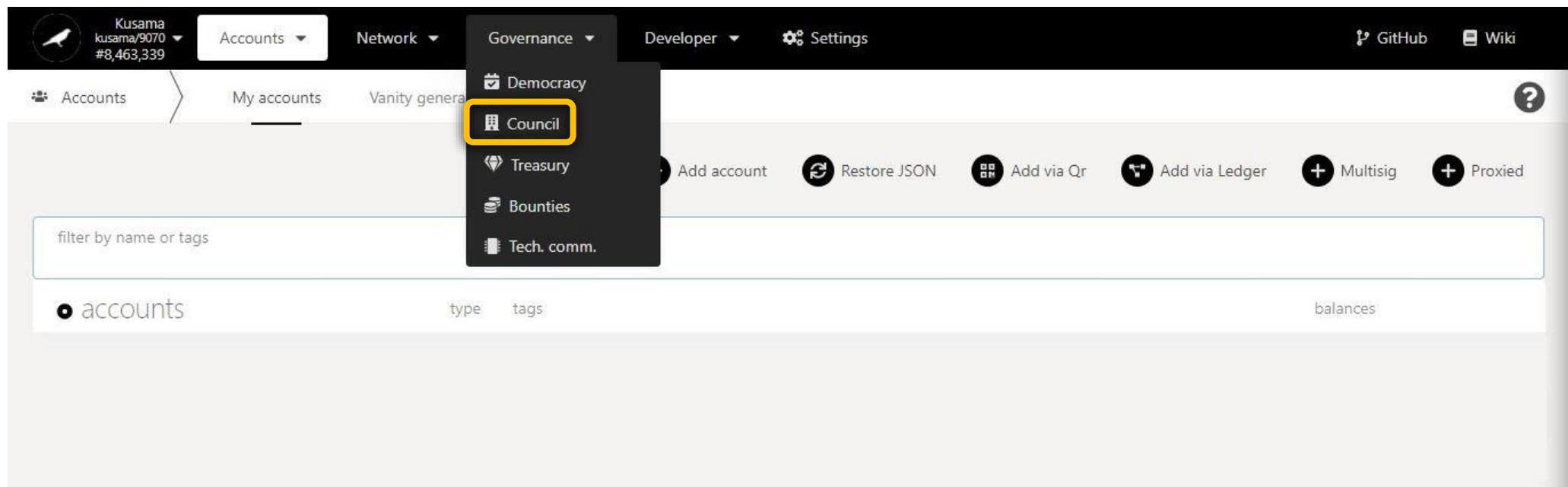
The call hash as calculated for this transaction

Cancel **Sign and Submit**

7. Click on Sign & submit to continue the procedure.



2. Council: Explore council-related activities.



a) Vote for council members, runners up and candidates.

The screenshot shows the Polkadot.js Governance interface with the following details:

- Header:** Kusama, kusama/9070, #8,200,195, Accounts, Network, Governance (selected), Developer, Settings, GitHub, Wiki.
- Breadcrumbs:** Council > Overview.
- Key Information:** Number of seats (19 / 19), Runners up (7 / 19), Candidates (0). Term progress: 1 day 13 hrs 30 s, 45%.
- Council Members:** A list of registered members with their account addresses: JACO, JAM, ROBOT HEART, JUTTA, AL SCIENTIST W3F, RTTI-5220.
- Council in detail:** Describes the council as the on-chain account of a group of network participants elected for 1 term (1 day). It also lists the council's responsibilities: referendum proposals, treasury proposals, tips, bounties, and technical committee memberships. It notes that any KSM holder can vote for registered councillors/runners up/candidates, requiring a voting deposit of 0.0675 KSM.
- Voting:** A callout box with the text "1. Click on vote." points to a "Vote" button next to a checked checkbox icon.
- Cumulative vote value:** A callout box describes the cumulative vote value received by each councillor/runner up.
- Backing and votes table:** A table showing the cumulative vote values for each councillor/runner up:

Backing	votes
111,640.6114 KSM	75
97,520.3107 KSM	45
97,171.4717 KSM	13
87,447.6048 KSM	24
87,016.4618 KSM	36
84,610.9768 KSM	44
- Callouts:**
 - A green arrow points from the "members" list to the "Councillor identity/account address." text.
 - A green arrow points from the "Backing and votes table" to the "Number of votes received by each councillor/runner up." text.
- Warning:** A red box at the bottom left contains the text "NEVER SEND YOUR KSM TO A COUNCILLOR'S ADDRESS!"

2. Follow on-screen instructions carefully.

The vote will be recorded for the selected account.

The value associated with this vote. The amount will be locked (not available for transfer) and used in all subsequent elections.

The votes for the members, runner-ups and candidates. These should be ordered based on your priority.

In calculating the election outcome, this prioritized vote ordering will be used to determine the final score for the candidates.

The amount will be reserved for the duration of your vote

3. Enter a KSM amount for your vote.

4. Click on 1-16 candidates' names to order them in your selection.

5. Click on Vote to continue the procedure.

The screenshot shows the Polkadot-JS Governance interface for the Kusama network. The user is currently in the 'Council' section. The top bar shows the account 'ANAEILLE LTD@KSM' with a balance of 1.0376 KSM. Below this, the 'vote for current candidates' section is displayed. It includes fields for 'voting account' (selected), 'vote value' (0.6000 KSM), and 'voting bond' (67.4133 KSM). A green arrow points to the 'vote value' field. The 'council candidates' list contains names like POLKAWORLD, DAN REECER, JACK, CHORUS ONE/3, KEVIN LI | EPC, and GAVIN - FIGMENT NETWORKS, each with a checked checkbox. A second green arrow points to the first candidate, POLKAWORLD. To the right, a 'my ordered votes' list shows the same candidates in a different order: RTTI-5220, CHEVDOR, JAM, SHAWN TABRIZI, and BRUNO | W3F. At the bottom right, there are buttons for 'Cancel', 'Unvote all', and 'Vote'. The 'Vote' button is circled in green. The entire interface is framed by a blue border with white text boxes containing the numbered steps and explanatory text.

Nature of the transaction.

The screenshot shows the Polkadot-JS extension interface for a Kusama chain. On the left, there's a sidebar with a 'Council' section showing 19 seats available. The main area is titled 'authorize transaction' and shows a transaction for 'phragmenElection.vote(votes, value)'. A green box highlights the fee information: 'Fees of 94.9990 micro KSM will be applied to the submission'. Below this, it says 'sending from my account' and lists 'ANAEILLE LTD@KSM'. There's also a note about not including a tip for the block author. The transaction hash is shown as '0xa41823f02dd3006854dbb1f411776129025f4ea67d9fdabde11adaa3acb152ea'. At the bottom, there's a 'Sign and Submit' button with a green circle around it, and a 'Cancel' button. A yellow box at the top right says 'Nature of the transaction.' with a green arrow pointing to the transaction details. A blue box at the bottom right says '7. Click on Sign & submit to continue the procedure.'

authorize transaction

Sending transaction `phragmenElection.vote(votes, value)`

Vote for a set of candidates for the upcoming round of election. This can be called to set the initial votes, or update already existing votes.

Fees of 94.9990 micro KSM will be applied to the submission

sending from my account
ANAEILLE LTD@KSM

Do not include a tip for the block author

call hash
0xa41823f02dd3006854dbb1f411776129025f4ea67d9fdabde11adaa3acb152ea

Sign and Submit

Cancel **Sign and Submit**

Nature of the transaction.

6. Check the transaction fees.

The details of the transaction including the type, the description (as available from the chain metadata) as well as any parameters and fee estimations (as available) for the call.

The sending account that will be used to send this transaction. Any applicable fees will be paid by this account.

Adding an optional tip to the transaction could allow for higher priority, especially when the chain is busy.

The call hash as calculated for this transaction

7. Click on Sign & submit to continue the procedure.

The screenshot shows the Polkadot.js extension interface. On the left, there's a sidebar with a Kusama account summary: Kusama/kusama/9070 #8,200,245. Below it, a 'Council' section shows 19 / 19 seats. A yellow box on the left says: "Summary of the transaction sent via the Polkadot-JS extension." A green arrow points from this box to the 'Accounts' dropdown menu in the top bar, which is currently set to 'polkadot[.js]'. The main window title is 'Transaction' and shows a transaction from 'Anaelle LTD@KSM' (HjcErRijmpoiBiKEHT3edPXM3NFycJogVwDPuByNe7hv9Ae) to 'phragmenElection.vote(signing)'. The transaction details are:

- from: https://polkadot.js.org/apps/#/accounts
- chain: Kusama
- version: 9070
- nonce: 94
- method: ► phragmenElection.vote(votes, value)
- info: ► Vote for a set of candidates for the upcoming round of election...
- lifetime: mortal, valid from 8,200,239 to 8,200,303

A green box highlights the password input field and the 'Remember my password for the next 15 minutes' checkbox. An orange button at the bottom is circled in green and labeled 'Sign the transaction'. To the right, a progress bar indicates '1 day 12 hrs 55 mins' with '46%' completed. A blue box contains the instruction: "8. Enter your account's password and tick the box to remember your password, if necessary." Another blue box at the bottom contains the instruction: "9. Click on Sign the transaction to complete the procedure."

Progress of the transaction.

1 day 12 hrs 55 mins 46%

Vote Submit candidacy

backing votes

111,640.6114 KSM	75
97,520.3107 KSM	45
97,171.4717 KSM	13
84,610.9768 KSM	44
83,813.2043 KSM	41
76,329.7010 KSM	37

Summary of the transaction sent via the Polkadot-JS extension.

PASSWORD FOR THIS ACCOUNT
•••••••••••••
 Remember my password for the next 15 minutes

Sign the transaction

8. Enter your account's password and tick the box to remember your password, if necessary.

9. Click on Sign the transaction to complete the procedure.

b) Clear council election votes.

The screenshot shows the Polkadot.js Governance interface for the Kusama network. The top navigation bar includes 'Accounts', 'Network', 'Governance' (selected), 'Developer', 'Settings', 'GitHub', and 'Wiki'. Below the navigation, a sidebar shows 'Council' (selected), 'Overview' (active), and 'Motions' (1). Key statistics are displayed: 19 / 19 seats, 7 / 19 runners up, 0 candidates, and term progress at 84% (1 day, 3 hrs 48 mins). A callout box with the text '1. Click on vote.' and an arrow points to the 'Vote' button next to the 'Submit candidacy' button. The main table lists members with their staking amounts and votes:

members	Backing	Votes
JACO	110,247.2670 KSM	77
ROBOT HEART	96,918.6914 KSM	15
JAM	95,960.5601 KSM	46
JUTTA	85,975.1974 KSM	25
RTTI-5220	83,467.7252 KSM	47

GUIDE TO POLKADOT-JS – PART IV: Governance

Version 2.0

The screenshot shows the Polkadot.js Governance interface. At the top, there are tabs for Accounts, Network, Governance (selected), Developers, and Settings. Below the tabs, it says "vote for current candidates". On the left, there's a sidebar with "Council" and "seats 19 / 19". The main area has sections for "voting account" (set to ANAELLE LTD@KSM), "vote value" (0.1000 KSM), and a "filter by name, address, or account index". Below these are two columns: "council candidates" (JACO, ROBOT HEART, JUTTA, AL SCIENTIST W3F, ACALA FOUNDATION) and "my ordered votes" (RTTI-5220, CHEVDOR, JAM, SHAWN TABRIZI, BRUNO | RMRK.APP). A "voting bond" of 67.4133 is shown. On the right, there are explanatory texts and a footer with "Cancel", "Unvote all" (circled in green), and "Vote".

vote for current candidates

voting account ? **ANAEILLE LTD@KSM** HjcErRijmpoiBiKEHT3edPXH3NFycJogVwDPuByN...
The vote will be recorded for the selected account.

vote value ? 0.1000 voting balance 0.2867 KSM KSM
The value associated with this vote. The amount will be locked (not available for transfer) and used in all subsequent elections.

Filter by name, address, or account index

council candidates my ordered votes

Backing	Votes
0 KSM	77
1 KSM	15
1 KSM	46
1 KSM	25
2 KSM	47
3 KSM	38
3 KSM	45
2 KSM	35

voting bond ? 67.4133 milli
The amount will be reserved for the duration of your vote

Cancel Unvote all Vote

2. Click on **Unvote all** to continue the procedure.

Nature of the transaction.

The screenshot shows the Polkadot-JS governance interface. On the left, there's a sidebar with icons for Kusama, Council, seats (19 / 19), members, JACO, RO, and JAM. The main area is titled "authorize transaction". It shows a transaction for "phragmenElection.removeVoter" with the status "queued". A green box highlights the message "Fees of 39.3329 micro KSM will be applied to the submission". Below this, it says "sending from my account ANAELLE LTD@KSM". There's a toggle switch for "Do not include a tip for the block author". A call hash is provided: "0x12d8978578c764ef6c58c6459c49e13f897410ff18b18bdb60efd6f69d4c1058". At the bottom, there are "Sign and Submit" and "Cancel" buttons, with "Sign and Submit" being circled in green. A blue box at the bottom right contains the instruction "4. Click on Sign & submit to continue the procedure.".

authorize transaction

Sending transaction phragmenElection.removeVoter()

Remove origin as a voter.

Fees of 39.3329 micro KSM will be applied to the submission

sending from my account ANAELLE LTD@KSM

HjcErRijmpoiBiKEHT3edPXH3NFycJogVwDPuByN...

Do not include a tip for the block author

call hash
0x12d8978578c764ef6c58c6459c49e13f897410ff18b18bdb60efd6f69d4c1058

Sign and Submit

Cancel  Sign and Submit

4. Click on Sign & submit to continue the procedure.

The screenshot shows the Polkadot-JS extension interface for the Kusama network. A yellow box on the left labeled "Summary of the transaction sent via the Polkadot-JS extension." highlights the transaction details. A green arrow points from this summary to the "Sign the transaction" button. Another green arrow points from the "Sign the transaction" button to a blue box containing step 6. A blue box on the right labeled "5. Enter your account's password and tick the box to remember your password, if necessary." highlights the password input field and the "Remember my password for the next 15 minutes" checkbox. A green arrow points from this box to a yellow box containing step 5. A yellow box at the top right labeled "Progress of the transaction." shows the transaction status as "phragmenElection.removeVoter signing" with a progress bar at 84% complete over 1 day and 3 hours 45 mins. It also includes "Vote" and "Submit candidacy" buttons.

Summary of the transaction sent via the Polkadot-JS extension.

from https://cloudflare-ipfs.com/ipns/dotapps.io/?
rpc=wss%3A%2F%2Fkusama.api.onfinality.io%2Fpublic-ws#/council
chain Kusama
version 9080
nonce 111
method ► phragmenElection.removeVoter()
info ► Remove `origin` as a voter.
lifetime mortal, valid from 8.680.936 to 8.681.000

PASSWORD FOR THIS ACCOUNT
.....
 Remember my password for the next 15 minutes

Sign the transaction

5. Enter your account's password and tick the box to remember your password, if necessary.

6. Click on Sign the transaction to complete the procedure.

Progress of the transaction.

phragmenElection.removeVoter signing
1 day 84%
3 hrs 45 mins

Vote Submit candidacy

Backing	Votes
110,247.2670 KSM	77
96,918.6914 KSM	15
95,960.5601 KSM	46
82,431.0263 KSM	38
79,221.0638 KSM	45
75,294.8522 KSM	35

c) Submit a candidacy for council.

Kusama
kusama/9070
#8,200,251

Accounts Network Governance Developer Settings

Council Overview Motions 1

seats runners up candidates
19 / 19 7 / 19 0

term progress
1 day
12 hrs 54 mins 46%

Council candidacy in detail:

A **council candidacy** is used to run a **campaign for election** as one of the 19 councillors or 19 runners up.

Any KSM holder can register to become a councillor/runner up and vote for him/herself. A candidacy deposit of 0.0034KSM is required and will be refunded once the candidacy is cleared.

1. Click on **Submit candidacy.**

Vote **Submit candidacy**

Backing	Votes
111,640,6114 KSM	75
97,520,3107 KSM	45
97,171,4717 KSM	13
87,447,6048 KSM	24
87,016,4618 KSM	36

GUIDE TO POLKADOT-JS – PART IV: Governance

Version 2.0

The screenshot shows the Polkadot.js extension interface for Kusama. At the top, it displays 'Kusama kusama/9070 #8,200,258'. Below this, the 'Council' section shows 'seats 19 / 19' and 'members'. The main area is titled 'submit your council candidacy'. It includes fields for 'candidate account' (ANAELE LTD@KSM) and 'candidacy bond' (3.3333 milli KSM). To the right, a blue box contains the instruction '2. Follow on-screen instructions carefully.' with two callout boxes. The first box, highlighted with a green border, states: 'This account will appear in the list of candidates. With enough votes in an election, it will become either a runner-up or a council member.' The second box states: 'The bond will be reserved for the duration of your candidacy and membership.' At the bottom right are 'Cancel' and 'Submit' buttons, with 'Submit' circled in green.

2. Follow on-screen instructions carefully.

This account will appear in the list of candidates. With enough votes in an election, it will become either a runner-up or a council member.

The bond will be reserved for the duration of your candidacy and membership.

Cancel Submit

3. Click on **Submit** to continue the procedure.

The screenshot shows the 'authorize transaction' screen for the same Kusama instance. It displays the transaction details: 'phragmenElection.submitCandidacy(candidate_count)' with a note 'Submit oneself for candidacy. A fixed amount of deposit is recorded.' A green box highlights the fee information: 'Fees of 39.6663 micro KSM will be applied to the submission'. To the right, a yellow box contains the instruction '4. Check the transaction fees.' with a callout box. The box states: 'Nature of the transaction.' followed by a green arrow pointing to 'phragmenElection.submitCandidacy queued'. It provides details about the transaction type, description, and fee estimations. Below this, another callout box shows the sending account information: 'sending from my account ANAELE LTD@KSM'. Further down, there are fields for 'call hash' (0xf3d7d91e7cfb2d484b5b78b40025a6ad90aa286a0851900b9a952393289b95bc) and a 'Sign and Submit' toggle switch. At the bottom right are 'Cancel' and 'Sign and Submit' buttons, with 'Sign and Submit' circled in green.

authorize transaction

Sending transaction phragmenElection.submitCandidacy(candidate_count)
Submit oneself for candidacy. A fixed amount of deposit is recorded.

Fees of 39.6663 micro KSM will be applied to the submission

Nature of the transaction.

phragmenElection.submitCandidacy queued

The details of the transaction including the type, the description (as available from the chain metadata) as well as fee estimations (as available) for the call.

sending from my account ANAELE LTD@KSM

Do not include a tip for the block author

call hash
0xf3d7d91e7cfb2d484b5b78b40025a6ad90aa286a0851900b9a952393289b95bc

Sign and Submit

Cancel Sign and Submit

5. Click on **Sign & submit** to continue the procedure.

The screenshot shows the Polkadot-JS extension interface for the Kusama network. A yellow box on the left contains the text: "Summary of the transaction sent via the Polkadot-JS extension." A green arrow points from this box to the "Accounts" dropdown menu. Another green arrow points from the "Sign the transaction" button to the "Sign the transaction" step in the guide below.

Transaction

Anaelle LTD@KSM

from https://polkadot.js.org/apps/#/accounts
chain Kusama
version 9070
nonce 94
method ► phragmenElection.submitCandidacy(candidate_count)
info ► Submit oneself for candidacy. A fixed amount of deposit is req...
lifetime mortal, valid from 8,200,263 to 8,200,327

PASSWORD FOR THIS ACCOUNT
Sign the transaction

Remember my password for the next 15 minutes

phragmenElection.submitCandidacy signing

Progress of the transaction.

1 day 46%
12 hrs 53 mins

Vote Submit candidacy

	Backing	Votes
111,640.6114 KSM	75	
97,520.3107 KSM	45	
97,171.4717 KSM	13	
84,610.9768 KSM	44	
83,813.2043 KSM	41	
76,329.7010 KSM	37	

6. Enter your account's password and tick the box to remember your password, if necessary.

7. Click on Sign the transaction to complete the procedure.

d) View council motions.

The screenshot shows the Polkadot.js Governance interface. At the top, there is a navigation bar with links for 'Accounts', 'Network', 'Governance' (which is currently selected), 'Developer', and 'Settings'. Below the navigation bar, there is a header with the network name 'Kusama' and account details 'kusama/9070 #8,467,828'. On the left, there is a sidebar with a 'Council' icon and three tabs: 'Overview' (selected), 'Motions' (highlighted with a green arrow and the number '1'), and 'Members'. The main content area displays the following information:

- seats**: 19 / 19
- runners up**: 7 / 19
- candidates**: 0
- term progress**: 1 day (22 hrs 57 mins)

Below this, there are two buttons: 'Vote' and 'Submit candidacy'. The 'Members' section is shown as a table:

members	backing	votes
JACO	110,846.6839 KSM	74
ROBOT HEART	96,875.3743 KSM	14
JAM	96,159.1499 KSM	44
JUTTA	86,677.2524 KSM	24
RTTI-5220	84,265.3891 KSM	43

Key information on motions: **proposal ID, nature of the proposal, voting minima for approval, and time left to approve/reject.**

motions	threshold	voting end	votes
332 treasury.approveProposal Approve a proposal. At a later time, the proposal will be allocated to the beneficiary and the original deposit will be returned.	12	2 days 20 hrs #8,509,030	Aye 1/12 ▾ <input checked="" type="checkbox"/> Vote

2. Click on the dropdown arrow to view the motion in detail.

Motion-related external links.

The screenshot shows the Polkadot.js governance interface. At the top, there's a navigation bar with tabs for 'Council', 'Overview', and 'Motions' (which is currently selected). Below the navigation, a yellow box highlights the text: 'Key information on motions: proposal ID, nature of the proposal, voting minima for approval, and time left to approve/reject.' A green box surrounds the first motion listed. This motion has a proposal ID of 332, is titled 'treasury.approveProposal', and describes it as 'Approve a proposal. At a later time, the proposal will be allocated to the beneficiary and the original deposit will be returned.' To the right of the motion details, there's a 'dropdown arrow' icon with a green arrow pointing to it, and a row of motion-related external links icons (cancel slashes, GitHub, Wiki) with a yellow box around them. A blue box contains the instruction: '2. Click on the dropdown arrow to view the motion in detail.'

GUIDE TO POLKADOT-JS – PART IV: Governance

Version 2.0

The screenshot shows the Polkadot.js Governance interface. At the top, there's a navigation bar with tabs for 'Council', 'Overview', and 'Motions' (which is currently selected, indicated by a bold underline and a small '1' in a circle). Below the navigation is a row of buttons: '+ Propose motion', '+ Propose external', and 'Cancel slashes'. The main area displays a table of motions. One specific motion is highlighted with a green border. This motion is titled 'treasury.approveProposal' and describes approving a proposal to allocate funds to a beneficiary. The motion details include:

proposal_id	threshold	voting end	votes
Compact<ProposalIndex> 100			
proposal: TreasuryProposal { proposer: ELkVhHcvaP9L43RK9SP6Wn3FBfusEN5EJe2gH7aA2ETiufP, value: 47.4199 KSM, beneficiary: E5S77xkFnRBq3qwDaRgdo1uuB1LXXy72RzsSBdQoLL3HTZJ, bond: 2.3709 KSM }	12	2 days 20 hrs #8,509,030	Aye 1/12 ▾
beneficiary E5S77xkFnRBq3qwDaRgdo1uuB1LXXy72...			Vote
payout 47.4199	KSM		

A green arrow points to the 'Aye 1/12 ▾' dropdown button, which is part of a callout box containing the instruction: '3. Click on the dropdown arrow to view current votes in detail.'

Motion in detail:

A **motion** is a vote created by the council to **approve or reject** referendum/treasury/bounty proposals.

It summarises information on the proposals (nature, proposer, value, bond paid, etc) that are stored on-chain.

GUIDE TO POLKADOT-JS – PART IV: Governance

Version 2.0

The screenshot shows the Polkadot.js Governance interface for the Kusama network. The top navigation bar includes 'Accounts', 'Network', 'Governance' (selected), 'Developer', 'Settings', 'GitHub', and 'Wiki'. Below the navigation, there are tabs for 'Council', 'Overview', and 'Motions' (selected). A '1' badge is visible next to the 'Motions' tab. On the right, buttons for 'Propose motion', 'Propose external', and 'Cancel slashes' are available.

The main area displays a single motion titled 'treasury.approveProposal'. The description states: 'Approve a proposal. At a later time, the proposal will be allocated to the beneficiary and the original deposit will be returned.' The motion details include:

- proposal_id:** Compact<ProposalIndex>
100
- proposal:** TreasuryProposal
{
proposer: ELkVhHcvaP9L43RK9SP6Wn3FBfusEN5EJe2gH7aA2ETiufP,
value: 47.4199 KSM,
beneficiary: E5S77xkFnRBq3qwDaRgdo1uuB1LXXy72RzsSBdQoLL3HTZJ,
bond: 2.3709 KSM
}
- beneficiary:** E5S77xkFnRBq3qwDaRgdo1uuB1LXXy72RzsSBdQoLL3HTZJ
- payout:** 47.4199 KSM

On the right side, the motion status is shown as 'Aye 1/12' with a timestamp of '2 days 20 hrs' and a balance of '#8,509,030'. A green checkmark icon is next to the identifier 'RTTI-5220'. A 'Vote' button with a checkmark is present, along with icons for 'Comment', 'Share', and 'Copy'. Two green arrows point upwards from a yellow callout box to the 'Polkassembly' icon and the 'Comment' icon.

Identity of the approving councillor(s).

4. Click on the Polkassembly icon to view the discussion thread.

Discussions On-chain ▾

Kusama ▾ anaelleltd ▾

5. Read the **information on the treasury proposal.**

#332 Open-Source DApp API: Milestone 2

Proposed

tldr: Second of three milestones in which SubQuery and Fearless Wallet create an open source API that should be able to provide different types of applications with a sufficient (but read-only) view of all key data within the Kusama chain. E.g. a developer could use it to create their own chain explorer without needing to directly query chain data

We believe that the DApps ecosystem will benefit from a common core API that each parachain can integrate (and extend) to index and expose their chain data for future consumer facing applications (e.g. a wallet, explorer, or other dApp).

There will be 3 individually proposed milestones delivering open source projects, packages, and associated learning material to help decentralised app developers start building applications on Polkadot/Kusama

Who
This is a joint proposal between two parties - SubQuery/OnFinality and Fearless Wallet/SORAMITSU

What

1. Create an Open-Source SubQuery Project for Common Data
2. Document and Publish Learning Material and Tutorials

Council Votes

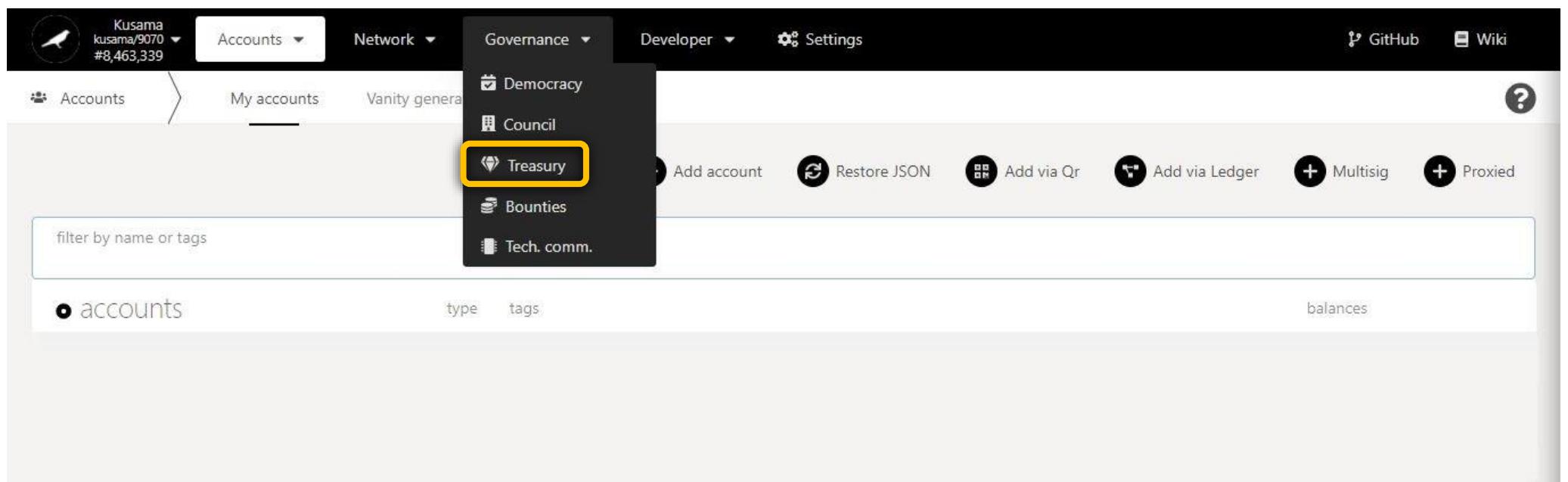
RTTI-5220 GLVe...F7wj Aye

Vote

No account found from the council :(

[Let me try still.](#)

3. Treasury: Explore treasury-related activities.



a) View and submit treasury proposals.

Basic information on technical words used in the TREASURY section.

Key information on treasury: **number of active & approved proposals, current pot & next burn amounts, time left before the next burn.**

proposals	beneficiary	payment	bond	
53 JOHNNYB	JOHNNYB	850.0000 KSM	42.5000 KSM	To council
76 HASHQUARK	HASHQUARK	523.0000 KSM	26.1500 KSM	To council

approved	beneficiary	payment	bond	
95 ONFINALITY/FRONTIER	E5577x...L3HTZJ	20.8800 KSM	1.0440 KSM	
96 OPENSQUARE	OPENSQUARE	79.4800 KSM	3.9740 KSM	

Proposal ID.

Proposer's Identity.

Treasury proposal in detail:

A **treasury proposal** is used to request funds from the Treasury pot.

The proposer needs to submit details of the proposal (i.e problem statement, proposed solutions, payout requested, milestones, etc) on a **public forum** for discussion.

A proposal bond of at least 5% of the requested payout is required and will only be refunded if the proposal is approved.

Beneficiary's account address.

Payout requested for the proposal and **deposit paid** by the proposer.

The screenshot shows the 'submit treasury proposal' form. It includes fields for 'submit with account' (ANAELE LTD@KSM), 'beneficiary' (ANAELE LTD@KSM), 'value' (5 KSM), 'proposal bond' (5.00%), and 'minimum bond' (66.666 milli KSM). A warning message at the bottom left states: '⚠️ Be aware that once submitted the proposal will be put to a council vote. If the proposal is rejected due to a lack of info, invalid requirements or non-benefit to the network as a whole, the full bond posted (as described above) will be lost.' At the bottom right are 'Cancel' and 'Submit proposal' buttons.

2. Follow on-screen instructions carefully.

This account will make the proposal and be responsible for the bond.

The beneficiary will receive the full amount if the proposal passes.

The value is the amount that is being asked for and that will be allocated to the beneficiary if the proposal is approved.

Of the beneficiary amount, at least 5.00% would need to be put up as collateral. The maximum of this and the minimum bond will be used to secure the proposal, refundable if it passes.

3. Double-check warning messages.

⚠️ Be aware that once submitted the proposal will be put to a council vote. If the proposal is rejected due to a lack of info, invalid requirements or non-benefit to the network as a whole, the full bond posted (as described above) will be lost.

4. Click on Submit proposal to continue the procedure.

Nature of the transaction.

The screenshot shows the Polkadot-JS extension interface for Kusama. The main title is "authorize transaction". Below it, the text reads: "Sending transaction treasury.proposeSpend(value, beneficiary) Put forward a suggestion for spending. A deposit proportional to the value is reserved and slashed if the proposal is rejected. It is returned once the proposal is awarded." A green box highlights the fee information: "Fees of 52.6661 micro KSM will be applied to the submission". To the right, a yellow box contains the transaction details: "treasury.proposeSpend queued". A green arrow points from the "Nature of the transaction." header to this box. The "Sign and Submit" button at the bottom right is circled in red.

5. Check the transaction fees.

Fees of 52.6661 micro KSM will be applied to the submission

sending from my account
ANAEILLE LTD@KSM

HjcErRijmpoiBiKEHT3edPXIM3NFycJogVwDPuByN...

Do not include a tip for the block author

call hash
0xd676169d5b150d3ec85cd5e5246aa4bddc4b751bf4d8e3e3b10e43865ec8d3d9

Sign and Submit

The details of the transaction including the type, the description (as available from the chain metadata) as well as any parameters and fee estimations (as available) for this type of call.

The sending account that will be used to send this transaction. Any applicable fees will be paid by this account.

Adding an optional tip to the transaction could allow for higher priority, especially when the chain is busy.

The call hash as calculated for this transaction

6. Click on Sign & submit to continue the procedure.

Cancel Sign and Submit

The screenshot shows the Polkadot-JS extension interface for the Kusama network. A yellow box on the left contains the text: "Summary of the transaction sent via the Polkadot-JS extension." A green arrow points from this text to the "treasury.proposeSpend signing" status bar at the top right. Another green arrow points from the "Sign the transaction" button in the bottom left to the "Progress of the transaction" bar on the right.

Summary of the transaction sent via the Polkadot-JS extension.

treasury.proposeSpend signing

Progress of the transaction.

6 days 12 hrs 49 mins 91%

Sign the transaction

7. Enter your account's password and tick the box to remember your password, if necessary.

8. Click on Sign the transaction to complete the procedure.

PASSWORD FOR THIS ACCOUNT

Approved
95
96

JOHNNYB HASHQUARK OPENQUARE

beneficiary	payment	bond	
JOHNNYB	850.0000 KSM	42.5000 KSM	To council
HASHQUARK	523.0000 KSM	26.1500 KSM	To council

b) View and submit treasury tips.

Key information on tips: **number of opened tips, current proposers, reasons for tips, and proposed tip payouts.**

1. Click on **Submit tip.** 

Number of tippers and proposed payout. 

tips	reason	Tippers	Tip
VEGAS_LIFE_MAIN/ksm	For a KSM 1KvP Leaderboard: http://vegas1kv.com/	(4) 2.1250 KSM	<input checked="" type="checkbox"/> <input type="checkbox"/>
E5qFqe...lwvtqrg	YouTube channel "Polkadot/Kusama what the FAQ" has many educational videos (e.g. https://www.youtube.com/watch?v=jHaGfO1WJ0Q)	(3) 18.0000 KSM	<input checked="" type="checkbox"/> <input type="checkbox"/>
GUpIE2...EWHNv6	Portuguese translation, subtitles and publication for Kusama vs Polkadot Especulação https://youtu.be/AOcdcw9uhFk	(2) 2.5000 KSM	<input checked="" type="checkbox"/> <input type="checkbox"/>
PLAYING WITH DUST	Creating phragviz, a visualizer to understand the phragmen election better: https://playing-with-dust.github.io/phragviz/	(2) 3.0000 KSM	<input checked="" type="checkbox"/> <input type="checkbox"/>
ALICE UND BOB/1	Kusama News - YouTube show: https://www.youtube.com/watch?v=YabQF8hiwhg	(3) 2.0000 KSM	<input checked="" type="checkbox"/> <input type="checkbox"/>
ALESSIO	For video tutorial on setting up Monitoring tools for Substrate validator in a k8s. https://youtu.be/6WdcC6o49QI		<input checked="" type="checkbox"/> <input type="checkbox"/>
EkQWFR...xXMrDr	For PolkaTalks (Programa 2) - Leilões de Parachains Kusama (1h about the PLOs)	(8) 5.0000 KSM	<input checked="" type="checkbox"/> <input type="checkbox"/>

Beneficiary's identity or account address. 

Treasury tips in detail:

A **treasury tip** is used to request funds from the Treasury pot and can be submitted:

- by yourself and for yourself (i.e the account address of the proposer and the beneficiary are the same).
- on behalf of someone else (i.e the account address of the proposer and the beneficiary are different). The proposer will **earn a finder's fee of 20% of the final tip amount**.

A tip deposit of about 0.0072KSM is required and will be refunded once the tip is closed.

Prepared by Anaelle LTD

GUIDE TO POLKADOT-JS – PART IV: Governance

Version 2.0

The screenshot shows the 'submit tip request' form in the Polkadot-JS Apps Treasury section. The form fields are as follows:

- submit with account**: ANAELLE LTD@KSM
- beneficiary**: ANAELLE LTD@KSM
- tip reason**: For creating a guide to Polkadot-JS Apps' Governance

On the right side of the form, there are two callout boxes with instructions:

- 2. Follow on-screen instructions carefully.**
 - Use this account to request the tip from. This can be a normal or council account.
 - The beneficiary will receive the tip as approved by council members.
 - A reason (to be stored-on-chain) as to why the recipient deserves a tip payout.
- 3. Click on Propose tip to continue the procedure.**

The 'Propose tip' button is highlighted with a green oval.

Nature of the transaction.

The screenshot shows the Polkadot.js extension interface for a Kusama node. The main title is "authorize transaction". Below it, the text reads: "Sending transaction tips.reportAwesome(reason, who) Report something reason that deserves a tip and claim any eventual the finder's fee." A green box highlights the text "Fees of 67.9993 micro KSM will be applied to the submission". To the right, a blue box contains the transaction details: "tips.reportAwesome queued". A green arrow points from the text "Nature of the transaction." to this box. The interface also includes sections for "sending from my account" (ANAELE LTD@KSM), "call hash" (0xc714d7ad81ac05fa0e0a343df16d3ba3f75ea5c397aa29f92327c496961a2ac4), and "Sign and Submit" (with a toggle switch). A blue box labeled "4. Check the transaction fees." is positioned above the "Sign and Submit" section. Another blue box labeled "5. Click on Sign & submit to continue the procedure." is positioned below it, pointing to the "Sign and Submit" button, which is circled in green.

authorize transaction

Sending transaction tips.reportAwesome(reason, who)
Report something reason that deserves a tip and claim any eventual the finder's fee.

Fees of 67.9993 micro KSM will be applied to the submission

4. Check the transaction fees.

tips.reportAwesome queued

5. Click on Sign & submit to continue the procedure.

The screenshot shows the Polkadot-JS extension interface for the Kusama network. A yellow box on the left highlights the "Summary of the transaction sent via the Polkadot-JS extension." A green arrow points from this summary to the password input field. Another green arrow points from the "Sign the transaction" button to a callout below it. A blue callout box contains the instructions: "6. Enter your account's password and tick the box to remember your password, if necessary." A yellow callout box on the right contains the instruction: "7. Click on Sign the transaction to complete the procedure." At the top right, a progress bar indicates "tips.reportAwesome signing" with a green arrow pointing to it, and a yellow box says "Progress of the transaction."

Summary of the transaction sent via the Polkadot-JS extension.

PASSWORD FOR THIS ACCOUNT
Remember my password for the next 15 minutes

Sign the transaction

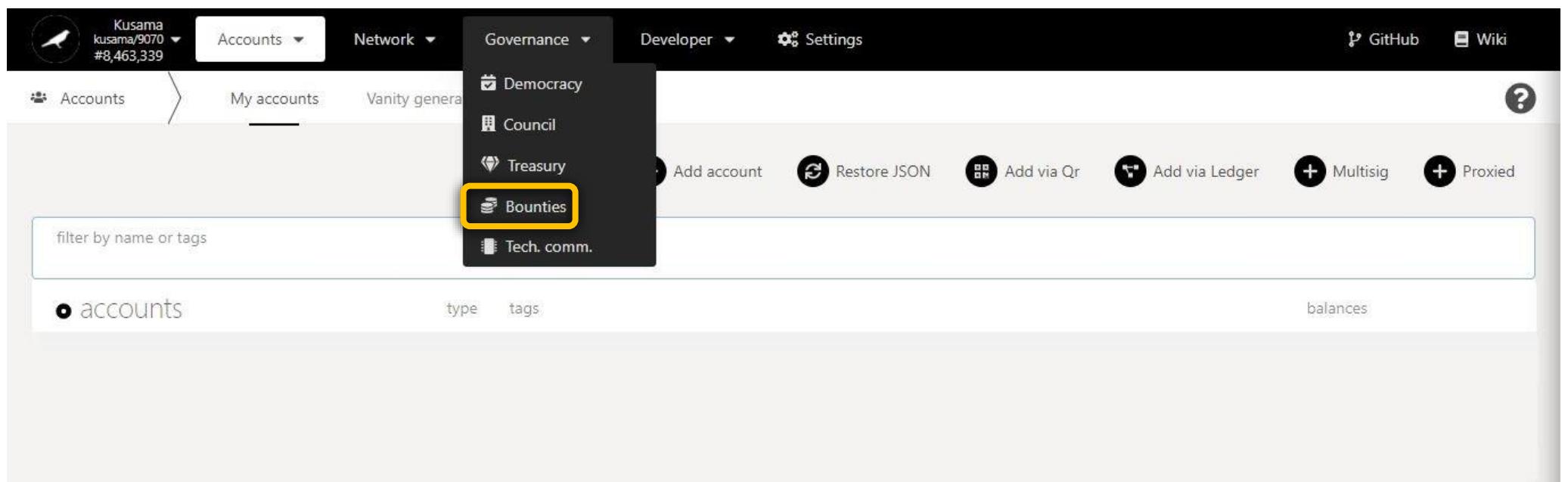
6. Enter your account's password and tick the box to remember your password, if necessary.

7. Click on Sign the transaction to complete the procedure.

tips.reportAwesome signing

Progress of the transaction.

4. Bounties: Explore bounty-related activities.



a) View active bounty proposals.

active past
2 4

active total
1,139 KSM

funding period
6 days
2 days 21 hrs 51%

Key information on Bounties: **number of active & approved bounties, value of active bounties, countdown to next bounty spending period.**

1. Click on the dropdown arrow for more details.				
bounties	value	curator	next action	
4 Council Alert App	Active ⓘ 28.1100 KSM	MARIO	63 days 5 hrs (update) #9,422,558	
3 @Polkadot/extension-inject on Mobile a...	Active ⓘ 1,111.0000 KSM	LITENTRY/HANWEN	33 days 52 mins (update) #8,987,636	

Bounty ID.

Bounty title.

Funding status & amounts.
Note: Bounties are implemented over an extended period.

Countdown to the next bounty milestone.

Bounty-related information.

GUIDE TO POLKADOT-JS – PART IV: Governance

Version 2.0

The screenshot shows the Polkadot.js Governance interface. At the top, there are navigation tabs: Accounts, Network, Governance (selected), Developer, and Settings. On the far right are links to GitHub and Wiki. Below the tabs, there's a header with the network name "Kusama", account "kusama/9070", and balance "#8,511,920". A sidebar on the left has "Bounties" and "Overview" buttons. The main area displays statistics: active bounties (2), past bounties (4), active total value (1,139 KSM), funding period (6 days, 2 days 21 hrs, 51%), and an "Add Bounty" button. A green arrow points to the "Add Bounty" button. Below this, a table lists bounties:

● bounties	value	curator	next action
4 Council Alert App	Active 28.1100 KSM	MARIO	63 days 5 hrs (update) #9,422,558
proposer RTTI-5220	bond 0.1949 KSM curator's fee 1.4000 KSM curator's deposit 0.7000 KSM		

A blue callout box with a green arrow points to the "curator" column of the first row, containing the name "MARIO". The callout text says: "2. Click on the **Polkassembly** icon to view the discussion thread."

Bounty proposal in detail:

A **bounty proposal** is used by council members to delegate the supervision of treasury proposals to expert curators.

The proposer needs to submit details of the proposal (i.e payouts, milestones, etc) on a public forum and **select a curator after approval**.

A proposal bond and a curator's deposit are both required and will only be refunded once all bounty milestones are completed.

The screenshot shows a web-based governance platform for Polkadot-JS. At the top, there's a navigation bar with the Polkassembly logo, 'Discussions', 'On-chain ▾', and user account information for 'Kusama' and 'anaelleltd'. A blue header bar contains the text '3. Read the information on the treasury proposal.' Below this, a green-bordered box contains the details of a specific proposal.

#4 Bounty Proposal: Kusama Council Alert App

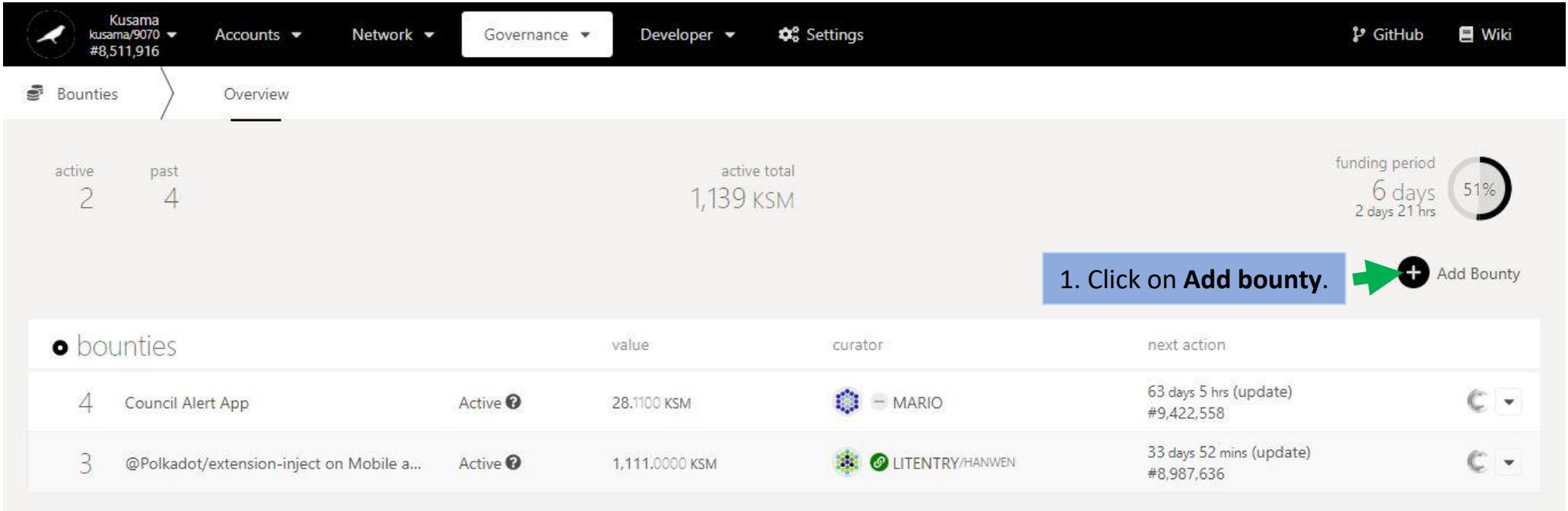
Extended

As we experience a continued growth in the Kusama ecosystem, it becomes imperative to find ways to communicate with stakeholders in an efficient manner, making sure all information is distributed for informed decisions to be taken, especially in the case of Council and Technical Committee members.

The goal of this proposal is to develop a basic app that listens to on-chain calls and notifies Council members when their review and vote is needed. An app to help councillors know once they are needed to push forward emergency mechanisms on-chain as well as other proposals would help mitigate the risk in the network. The app:

- would connect to rpc node;
- would connect with different networks;
- would notify about events;
- would watch wallet accounts and tell the user when pending;
- would have different alert levels;
- would recognise a "batcall" remark from council or TC members, with a `system.remark "alertCouncil"` that would trigger an alarm.

b) Submit bounty proposals.

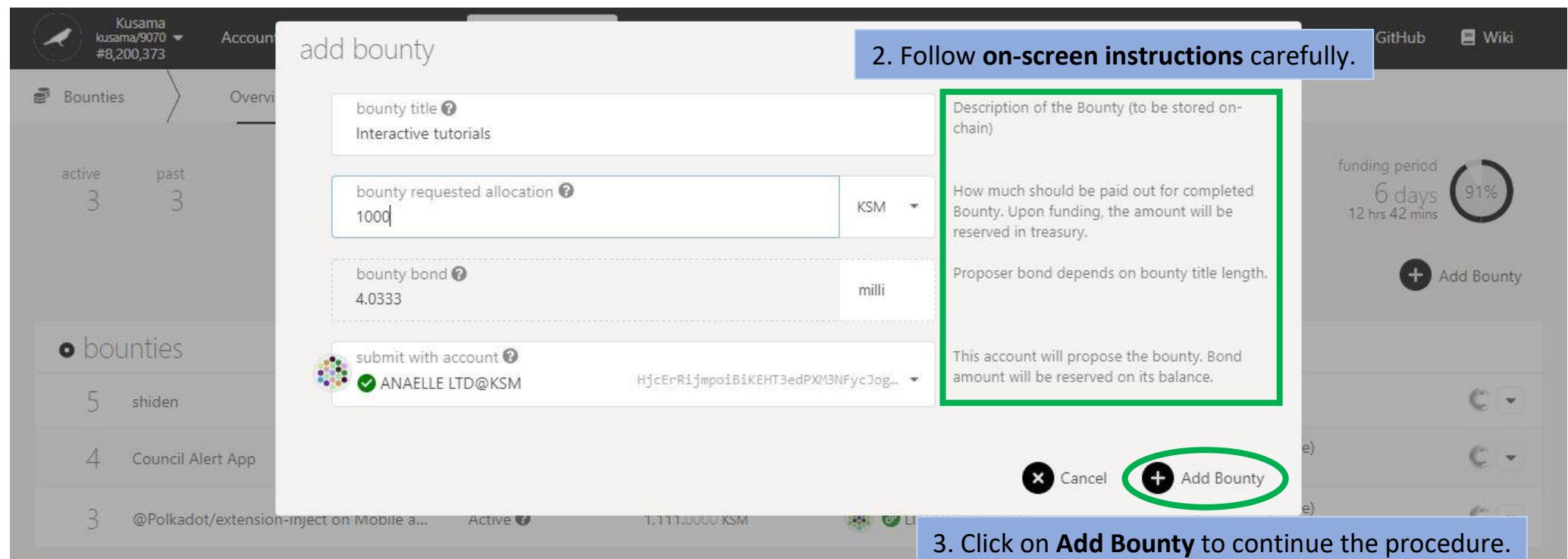


The screenshot shows the Polkadot.js Governance interface with the "Bounties" tab selected. At the top, there are tabs for "Overview" and "Bounties". Below the tabs, it shows "active" (2) and "past" (4) bounties. The "active total" is 1,139 KSM. To the right, a circular progress bar indicates a "funding period" of 6 days (51% complete). A callout box with a green arrow points to the "Add Bounty" button, which is located at the bottom right of the main content area.

● bounties	value	curator	next action
4 Council Alert App	Active ? 28.1100 KSM	MARIO	63 days 5 hrs (update) #9,422,558
3 @Polkadot/extension-inject on Mobile a...	Active ? 1,111.0000 KSM	LITENTRY/HANWEN	33 days 52 mins (update) #8,987,636

GUIDE TO POLKADOT-JS – PART IV: Governance

Version 2.0



Nature of the transaction.

The screenshot shows the Polkadot-JS extension interface for Kusama. The top bar displays the chain name 'kusama/9070 #8,200,381'. The main window title is 'authorize transaction' under the 'Bounties' tab. A green box highlights the transaction details: 'Fees of 49.3328 micro KSM will be applied to the submission'. To the right, a blue box contains the instruction '4. Check the transaction fees.' Below this, another blue box contains the instruction '5. Click on Sign & submit to continue the procedure.' A green arrow points from the text 'Nature of the transaction.' to the transaction details section. A green oval highlights the 'Sign and Submit' button at the bottom right of the transaction fees section.

authorize transaction

Sending transaction bounties.proposeBounty(value, description)

Propose a new bounty.

Fees of 49.3328 micro KSM will be applied to the submission

sending from my account ANAELLE LTD@KSM

HjcErRijmpoiBiKEHT3edPXl3NFycJogVwDPuByN...

Do not include a tip for the block author

call hash
0x8438f2687003586eec6b8a16980a706d69f8df5c1dd186b0af71c4e76f3555fc

Sign and Submit

bounties.proposeBounty queued

The details of the transaction including the type, the description (as available from the chain metadata) as well as fee estimations (as available) for the call.

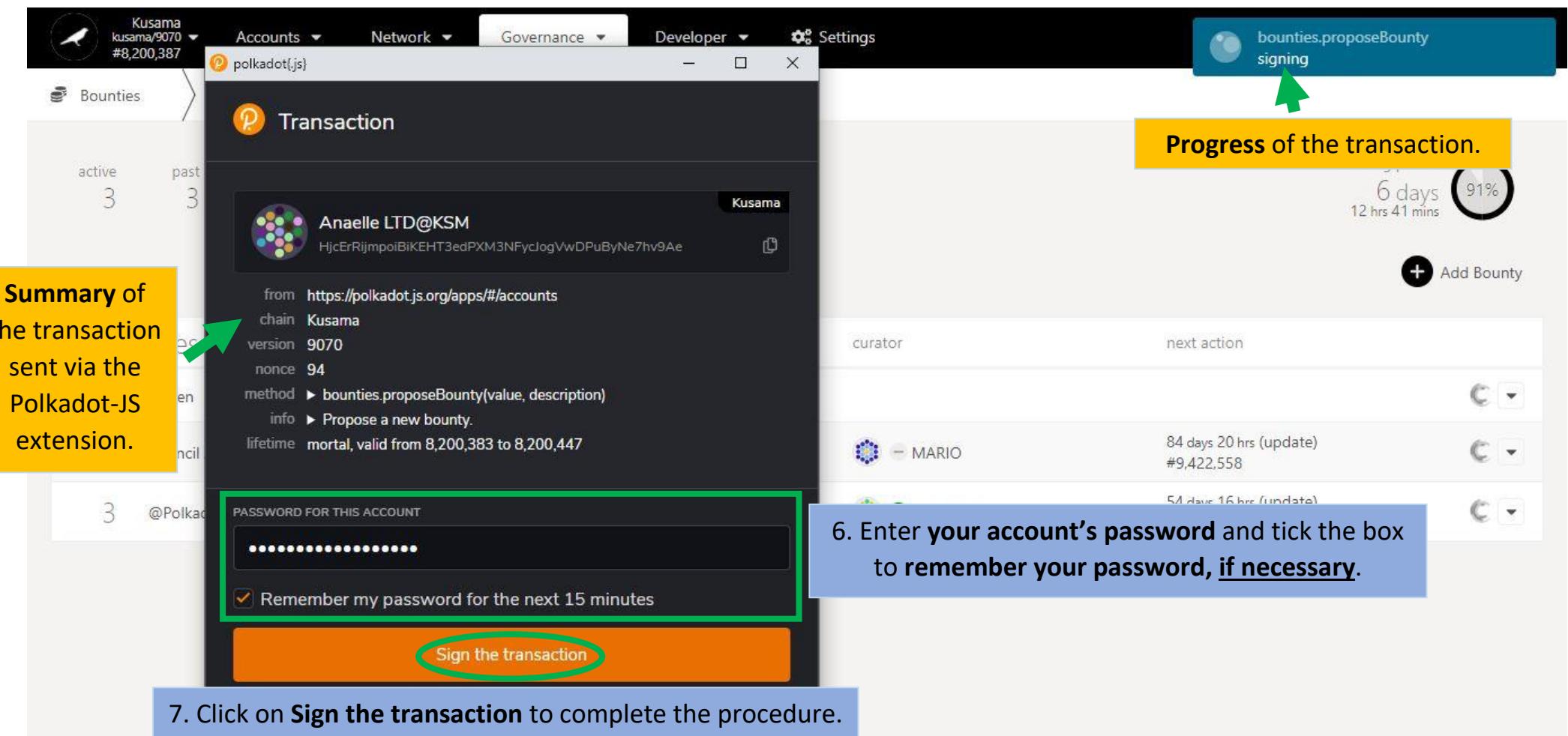
The sending account that will be used to send this transaction. Any applicable fees will be paid by this account.

Adding an optional tip to the transaction could allow for higher priority, especially when the chain is busy.

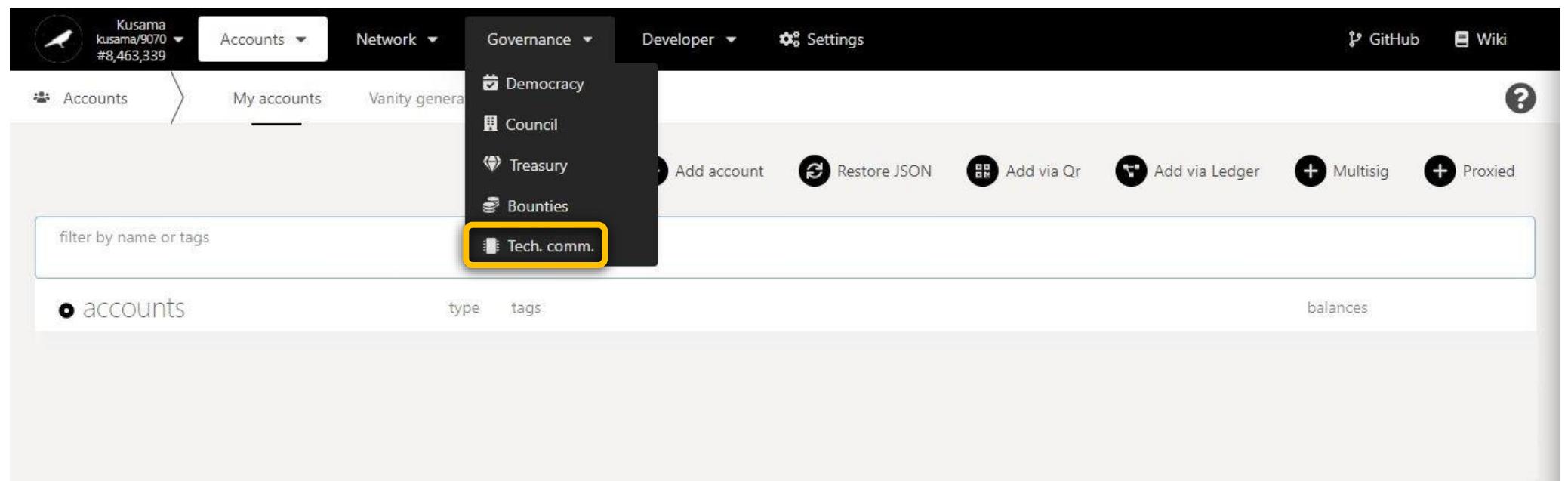
The call hash as calculated for this transaction

Cancel **Sign and Submit**

5. Click on **Sign & submit** to continue the procedure.



5. Technical Committee: Explore technical committee-related activities.



a) View technical committee members.

The screenshot shows the Polkadot.js Governance interface. At the top, there is a navigation bar with links for 'Accounts', 'Network', 'Governance' (which is currently selected), 'Developer', and 'Settings'. Below the navigation bar, there is a yellow banner with the text 'Key information on Tech committee: members, number of active & approved proposals.' On the left side, there is a sidebar with a tree view showing 'Tech. comm.', 'Overview', and 'Proposals (1)'. The main content area has a green header bar with the text 'members' and the value '3'. To the right of this, there are two columns: 'proposals' (value '1') and 'total' (value '84'). Below this, there is a table with three rows, each showing a team name and its corresponding icon. To the right of the table, there is a yellow box containing text about the Technical Committee members. The text states: 'Technical committee members are chosen within teams that are actively developing Relay chains runtime implementations. They are installed by the Council.'

members	proposals	total
3	1	84

members

PARITY/TECHNICAL 2

PARITY/TECHNICAL 1

WEB3 FOUNDATION/TECHNICAL

Technical committee in detail:

Technical committee members are chosen within teams that are actively developing **Relay chains runtime implementations**. They are installed by the Council.

b) View active technical committee proposals.

The screenshot shows the Polkadot.js Governance interface. At the top, there is a navigation bar with tabs for 'Accounts', 'Network', 'Governance' (which is currently selected), 'Developer', and 'Settings'. Below the navigation bar, there is a breadcrumb navigation with 'Tech. comm.' and 'Proposals (1)'. A yellow banner at the top of the main content area states: 'Key information on TC proposals: proposal ID, nature of the proposal, proposer, and time left to approve/reject.' On the right side of the banner is a '+ Submit proposal' button. The main content area displays a table of proposals. The first proposal, numbered 83, is highlighted with a green border and a green arrow pointing to its dropdown arrow. The proposal details are as follows:

proposal	threshold	voting end	aye	nay
democracy.fastTrack Schedule the currently externally-proposed majority-carries referendum to be tabled immediately. If there is no externally-proposed referendum currently, or if there is one but it is not a majority-carries referendum then it fails.	1/2	2 days 21 hrs #8,553,735	PARITY/TECHNICAL 2	<input checked="" type="checkbox"/> Vote

1. Click on the **dropdown arrow**
to view the proposal in detail.

GUIDE TO POLKADOT-JS – PART IV: Governance

Version 2.0

The screenshot shows the Polkadot.js governance interface. At the top, there's a navigation bar with tabs for 'Accounts', 'Network', 'Governance' (which is active), 'Developer', and 'Settings'. Below the navigation bar, there's a header with a profile icon for 'kusama/9070 #8,511,904', a 'GitHub' link, and a 'Wiki' link.

The main content area has a breadcrumb navigation: 'Tech. comm.' → 'Overview' → 'Proposals (1)'. On the right side of this breadcrumb, there's a 'Submit proposal' button. Below the breadcrumb, there's a section titled 'proposals' with a green border around it. This section contains a proposal for 'democracy.fastTrack'. The proposal details are as follows:

- proposal_hash:** Hash 0x23462d768da9eec611877a69db768e72b94e41e89d760bebf6a090508af131
- voting_period:** BlockNumber 43,200
- delay:** BlockNumber 600
- proposal hash:** 0x240654828d64b74064ddb829084a7109c4451c8035211a644f2f7fe1ed84aa7a

To the right of this proposal, there's a yellow box containing the text: 'TC proposal in detail: A TC proposal is used to schedule emergency referenda for bug fixes and fast-track runtime upgrades.'

Below the proposal details, there are some statistics: 'threshold' (1/2), 'voting end' (2 days 21 hrs), 'aye' (#8,553,735), and 'nay' (0). There's also a 'Vote' button with a checked checkbox.