User Story Inventory

# Document Summary

Upon socialising of this Inventory, and eventual agreement, we will progress to detailing how the Acceptance will be accomplished and signoff achieved, the ‘Accepted’ column will be marked Yes once consensus is reached.

# Change Management

Upon agreement on the User Story Inventory, a suitable level of Change Management will be applied, this will enable us to phase-deliver Governance on Cardano and to fulfil the ongoing Ecosystem and Community needs. Additional User Stories will be proposed, reviewed, approved and accepted using

# Source Areas Under Consideration

This section shows the areas or sources thus far

| **Source** | **Outline Contribution** | **State** | **Key Contract** |
| --- | --- | --- | --- |
| Voltaire | Use Cases for Governance, voting, delegation | Initial Capture Complete | Outreach in progress |
| CIP-1694 | Blockchain level User Stories on voting, delegation, identity, consensus | Initial Capture Complete | Outreach in progress |
| Smart Contracts | Plutus v3 and BLS | Initial Capture Complete | Outreach in progress |
| CLI-API | Interface outline and definition | Initial Capture Complete | Outreach in progress |
| Sidechains | Interaction with BLS primitives | Outreach in progress | Outreach in progress |
| DApps | To be defined and elaborated | To be defined | Outreach in progress |
| Exchanges | To be defined and elaborated | To be defined | Outreach in progress |
| Other | To be defined and elaborated | To be defined | Outreach in progress |

# User Story (Voltaire, API, CIP-1694, Community et al)

| **UID** | **User Story** | **Functional Requirement** | **Acceptance Criteria** | **Link** | **Accepted** | **Source** | **Enabler(Y,N)** |
| --- | --- | --- | --- | --- | --- | --- | --- |
| CH.VO1 | As a DRep or Ada Holder **I want to connect my wallet to GovTool** so that I can post transactions on-chain | Connect with multiple stake key wallet | Given I am on the homepage    And my wallet is not connected.    When I click the Connect Wallet button    And select (one of) my CIP-95 compatible wallet(s) with multiple stake keys  And select from a list which stake key I wish to connect with    Then the wallet will prompt me to connect and I can connect to GovTool with it on the selected stake key. |  |  | Voltaire | N |
| Given I am on the homepage    And my wallet is not connected    When I click the Connect Wallet button    Then I am not shown any non CIP-95 compatible wallets. |  |  | Voltaire | N |
| Given I am on the homepage    And my wallet is not connected    When I click the Connect Wallet button and select a CIP-95 compliant, multiple stake key wallet, containing zero ADA (or tADA for SanchoNet)    Then a modal window opens showing my CIP-95 compatible wallets  When I select a wallet with multiple stake keys from this list  Then I am asked which stake key I wish to connect with  When I select a stake key  Then the wallet opens and I can connect with it on the selected stake key. |  |  | Voltaire | N |
| Given i am on the homepage without my wallet connected    When I click the Connect Wallet button and select a CIP-95 compliant, single stake key wallet, containing more than zero ADA (or tADA for SanchoNet)    Then a modal window opens showing my CIP-95 compatible wallets  When I select a wallet with multiple stake keys from this list  Then I am asked which stake key I wish to connect with  When I select a stake key  Then the wallet opens and I can connect with it on the selected stake key. |  |  | Voltaire | N |
| Connect with single stake key wallet | Given I am on the homepage with no wallet connected  When I click the Connect Wallet button and select a CIP-95 compliant single stake key wallet  Then my wallet appears and I can connect with it |  |  | Voltaire | N |
| Given I am on the homepage without my wallet connected    When I click the Connect Wallet button    Then I am not shown any non CIP-95 compatible wallets. |  |  | Voltaire | N |
| Given I am on the homepage without my wallet connected    When I click the Connect Wallet button and select a CIP-95 compliant, single stake key wallet, containing zero ADA (or tADA for SanchoNet)  Then my wallet appears and I can connect with it |  |  | Voltaire | N |
| Given I am on the homepage without my wallet connected    When I click the Connect Wallet button and select a CIP-95 compliant, single stake key wallet, containing more than zero ADA (or tADA for SanchoNet)  Then my wallet appears and I can connect with it |  |  | Voltaire | N |
| Disconnect wallet | Given that I am on the dashboard with my wallet connected  If I click the Disconnect button  Then my wallet is disconnected from GovTool and I am redirected to the homepage. |  |  | Voltaire | N |
| Given that I am connected to GovTool with my wallet  When I disconnect  Then I will be redirected to the homepage, and will not have access to delegation or voting features. |  |  | Voltaire | N |
| Check the wallet is on the correct network | Given I am on the homepage  When I compare the networkId with the environment value set on the deployment for the network.    Then if there are exceptions raised, fail the test.    If no exceptions, connect the wallet to the network (pass) |  |  | Voltaire | N |
| CH.VO2 | As an Ada Holder **I want to** **delegate my voting power** to a DRep so that I can claim my staking rewards | Delegate to DRep ID | Given that I have my wallet connected, and I am on the Delegate to DRep page  When I select the delegate to DRep ID option and I enter a DRep ID which has not been registered and I press delegate  Then I will be presented with an error message explaining that the DRep ID was not found. |  |  | Voltaire | N |
| Given that I have my wallet connected, and I am on the delegate to DRep page,    When I choose the Delegate to DRep ID option and I enter a registered DRep ID and I press the Delegate button    Then I am able to delegate to that DRep ID via my connected wallet |  |  | Voltaire | N |
| Given that I have connected to GovTool with zero\* ADA (or tADA in the case of SanchoNet)  When I choose the Delegate to DRep ID option and I enter a registered DRep ID and I press the Delegate button    Then I am presented with a warning message and cannot proceed with delegation.  \*or at least a number below transaction costs |  |  | Voltaire | N |
| Access Delegate to DRep page | Given that I do not have a compatible wallet connected to GovTool  When I attempt to visit the URL of the Delegate to DRep page  Then I am redirected to the homepage |  |  | Voltaire | N |
| Given that I have a compatible wallet connected to GovTool and I am looking at the dashboard  When I click on the the Delegate button (or Change Delegation button if you are already registered)  Then I am shown the Delegate to DRep page |  |  | Voltaire | N |
| Verify DRep behaviour in connected state | Given that I'm not connected to the wallet    When I visit the DRep delegation page, and I click the delegate-connect-wallet-button    Then the connect your wallet-modal is visible |  |  | Voltaire | N |
| Verify DRep behaviour in disconnected state | Given that I have a preset DRep wallet loaded    Then Delegate button is clicked  Then it is expected that delegation options card is visible  delegate to myself is expected to be visible  Then Other options is clicked  Expect that signal no confidence card and vote abstain cards are visible  Next, delegate to dRep card is clicked, followed by next step button  Then expected that dRep ID input is visible along with delegate button |  |  | Voltaire | N |
| Delegate to myself | Given that I am a registered DRep who is connected to GovTool with my wallet, and I am on the Delegate to DRep page  When I choose the Delegate to DRep ID option and I enter my own DRep ID and I press the Delegate button    Then I am able to delegate to myself via my connected wallet |  |  | Voltaire | N |
| Given that I am a registered DRep who is connected to GovTool with my wallet, and I am on the Delegate to DRep page  When I select the Delegate to Myself option and press the Delegate button  Then I will be able to send a transaction to delegate to myself via my wallet |  |  | Voltaire | N |
| Given that I am not a registered DRep, and I am connected to GovTool with my wallet,    When I am on the Delegate to DRep page    I cannot see a Delegate to Myself option |  |  | Voltaire | N |
| Change my DRep delegation | Given that I am I am already delegated to a DRep  When I look at the dashboard  GovTool will know that I am delegated and it will invite me to “change my delegation” rather than to delegate. |  |  | Voltaire | N |
| Given that I am already delegated  When I go to change my delegation  I can delegate to any registered DRep, If I am delegated to myself then the option to “delegate to myself” will not be shown, If I am delegated to a specific predefined DRep then this predefined option will not be shown. |  |  | Voltaire | N |
| Check the validity of a DRep ID | Given that I have selected the “delegate to a DRep ID” option in the delegation user journey.  When I enter anything in the DRep ID input box that is not a registered DRep ID.   Then I will not be able to delegate to this DRep ID and will get a warning message. |  |  | Voltaire | N |
| Delegate to Abstain | Given that I am a DRep    When I delegate using the “delegate to abstain” feature    Then it will only delegate my own lovelace’s voting power to Abstain and NOT the voting power (if any) that has been delegated to me by others. I will be notified that my delegation transaction was sent. |  |  | Voltaire |  |
| Given that I am not a DRep    When I delegate using the “delegate to abstain” feature    Then it will delegate any voting power I have to Abstain. I will be notified that my delegation transaction was sent. |  |  |  |  |
| Delegate to No-Confidence | Given that I am a DRep    When I delegate using the “delegate to no-confidence” feature    Then it will only delegate my own lovelace’s voting power to No-Confidence and NOT the voting power (if any) that has been delegated to me by others. I will be notified that my delegation transaction was sent. |  |  | Voltaire |  |
| Given that I am not a DRep    When I delegate using the “delegate to no-confidence” feature    Then it will delegate any voting power I have to No-Confi. I will be notified that my delegation transaction was sent. |  |  | Voltaire |  |
|  |  | Guardrails for Voltaire | This work is in progress |  |  | Voltaire |  |
| CH.VO3 | **As a DRep I want to register** so that I can vote on governance actions | Register as a DRep | Given that I am connected to GovTool with a compatible wallet  When I go through the DRep registration process, and do not include a metadata anchor  Then I can register as a DRep via my wallet (because metadata anchors are optional) |  |  | Voltaire | N |
| Given that I am connected to GovTool with a compatible wallet  When I go through the DRep registration process, and include metadata anchor information in the wrong format  Then I will not be able to progress further in the process and I will be told that it is because the format is incorrect. |  |  | Voltaire | N |
| Given that I am connected to GovTool with a compatible wallet  When I go through the DRep registration process, and include metadata anchor information in the correct format  Then I will be able to register as a DRep via my wallet, GovTool will include the metadata anchor in the registration certificate transaction. |  |  | Voltaire | N |
| Confirm transaction has been sent | Given that I have gone through the DRep registration process  When I press the button on my wallet to submit the transaction  Then I will receive a confirmation message from GovTool that will include a link to the transaction in a block explorer. |  |  | Voltaire | N |
| Status of transaction is displayed | Given that I have just submitted a DRep registration transaction, and I am looking at the dashboard    When the registration has not yet been confirmed by the blockchain,  Then the registration status will show as “In Progress” until it is confirmed. |  |  | Voltaire | N |
| CH.VO4 | **As a DRep I want to vote** so that I can fulfil my role | Should be able to access the governance actions page as a DRep with my wallet connected | Given that I am a DRep and I am connected to GovTool    When I visit the url of the governance actions page    Then the governance actions page is displayed |  |  | Voltaire | N |
| Given that I am a DRep and connected to GovTool  When I look at the governance actions page    Then my voting power is displayed |  |  |
| Given that I am a DRep and Connected to GovTool, and I am on the governance actions page  When I click Disconnect Wallet    Then my wallet is disconnected and I am redirected to the same page, but without the DRep functionality (i.e. ability to vote) |  |  |
| Given that I am a DRep and I am on the governance actions page  When I click on the “view proposal details” button  Then I will be shown the page for that individual governance action and be able to view its details |  |  |
| A DRep should be able to vote on a live governance action | Given that I am a DRep    When I look at the details page of an individual governance action    Then I can see how many votes the governance action currently has for, against and abstaining. |  |  | Voltaire | N |
| Given that I am a DRep    When I look at the details page of an individual governance action    Then there are buttons allowing me to vote for, against or abstain. |  |  |
| Given that I am a DRep, on the details page of an individual governance action  When I select yes/ no/ abstain, and click vote  Then I can sign & submit this vote via my wallet |  |  |
| Given that I am a DRep  When I have submitted a vote  Then Immediately after this GovTool will display a message informing me that my transaction has been sent and providing me with a link to a block explorer where I can view the transaction |  |  |
| People without the (t)ADA needed to pay for voting transactions should not be able to submit a voting transaction | Given I have less Lovelace in my wallet than a transaction costs    When I attempt to vote    The GovTool will tell me that there is an error |  |  | Voltaire | N |
| People without their wallet connected or who do have their wallet connected but have not registered as DReps should not be able to vote | Given that I do not have a wallet connected to GovTool    When I visit the details of a governance action    Then I am not shown a vote button. |  |  | Voltaire | N |
| No one should be able to vote on a governance action that has expired, or been ratified, or enacted. | Given that I am on the governance action page  When I examine the governance actions  None of the governance actions shown on the page have expired or been ratified or enacted. |  |  | Voltaire | N |
| A DRep should be able to change their vote | Given that I am a DRep and I have already voted on a given governance action  When I submit a different vote for the same transaction within the same snapshot    Then the most recent vote will be counted. |  |  | Voltaire | N |
| Given that I have already cast a vote on a governance action  When I examine this specific governance action’s page  Then I can see that I have already voted and what my most recent vote was |  |  |  | N |
| Given that I have already cast a vote on a given governance action  When I examine this specific governance action’s page  Then instead of seeing a “vote” button I should see a “change vote” button |  |  | Voltaire | N |
| Only the votes of participants who are still DReps at the relevant epoch boundary will be accepted | Given that I am a DRep and I vote yes or abstain on a live governance action.    At the epoch boundary    My votes are counted. |  |  | Voltaire | N |
| Given that I was a DRep that voted yes or abstain on a live governance action but then retired.    At the next epoch boundary    My votes will not be counted towards the total tally of DRep votes. |  |  | Voltaire | N |
| DReps can attach a metadata anchor to their votes | Given that I have chosen how to vote on the UI of a governance action’s details    When I add a metadata anchor to the UI also and click the vote button    Then the resulting transaction will include my metadata anchor |  |  | Voltaire | N |
| CH.VO5 | **As a DRep I want to retire** so that I can reclaim my DRep Deposit | Only a user who is registered as a DRep can retire | Given that I am not registered as a DRep,    When I look for a retirement option in GovTool    Then there is none. |  |  | Voltaire | N |
| Given that I am registered as a DRep,    When I look for a retirement option in GovTool there is one. And when I choose this option    Then my wallet opens and I can sign a retirement action which is registered on-chain. |  |  |
| When I retire I get my deposit back | Given that I am a DRep    When I register a valid retirement transaction on chain    then my DRep registration deposit will be returned to me. |  |  |  | N |
| Only a user who has the wallet that they registered as a DRep with can retire. | Given that I am not connected to GovTool    When I look at the homepage    Then I will not see an option to retire |  |  | Voltaire | N |
| Given that I am connected to GovTool with an account that is not associated with a registered DRep certificate    When I look at the homepage    Then I see an option to register as a DRep |  |  | Voltaire | N |
| Given that I am a registered DRep with my wallet account connected    When I click the retire as a DRep option on the homepage and then send the retirement transaction with my wallet    Then the blockchain will register my retirement certificate, and I will be retired. |  |  | Voltaire | N |
| CH.VO6 | **As a DRep I want to update my details** so that I can better advertise myself to Ada Holders | A DRep can update their registration after registering | Given that I am a DRep and am connected to GovTool and am on the dashboard.    Then when I click the “change metadata” button on the DRep tab.    Then I am directed to update my metadata and can submit a DRep update certificate to register this on-chain. |  |  | Voltaire | N |
| CH.VO7 | As any user **I want to view governance actions** so I can see what is being proposed | Anyone should be able to access the governance actions page without a wallet connected | Given that I am on the GovTool homepage,    When I click the “Governance actions” in the topbar    Then I am sent to the governance actions page. |  |  | Voltaire | N |
| I can see all live governance actions | Given that I am on the Governance Actions page    When I review the governance actions available to view on the page,    Then all of the non expired/ ratified/enacted governance actions. |  |  | Voltaire | N |
| Anyone with a CIP-95 compatible wallet connected should be able to access the governance actions page | Given that I am on the GovTool dashboard,    When I click the View Governance Actions tab, or “Governance actions” in the sidebar    Then I am sent to the governance actions page. |  |  | Voltaire | N |
| Should be able to see if a governance action has been accepted or rejected by the Constitutional Committee | Given that I am looking at an individual governance action,    When I look at how many votes it has,    Then it shows the number of CC votes and whether this is acceptance/veto or neither. |  |  | Voltaire | N |
| Should be able to view relevant information about governance actions | Given that I am looking at a Treasury Withdrawal governance action    Then I can see the amount of ADA that the proposal submitter wants to withdraw, and the address that they want to withdraw it to. |  |  | Voltaire | N |
| Given that I am looking at a Proposal Parameter Change governance action,    Then I can see the parameter(s) that the Proposal Submitter is proposing to change along with what the current values are, and what he wants to change them to. |  |  |
| Given that I am looking at a Constitutional Committee Update governance action,    Then I can see the following (where applicable): - Old Committee Member Cold Key Hash to be removed - Float, (a rational number in the range from 0 to 1 inclusive. Of course if all you have in a tool is Floats, than that is what it will have to be) - Map of committee cold key credentials that will be added to the committee with absolute epoch number number when they will expire |  |  |
| Given that I am looking at an Update to the Constitution governance action    Then I should be able to see: - The Constitution URL - The Constitution hash - The proposal policy script (if provided when the governance action was submitted) |  |  |
| Given that I am looking at a Hard Fork Initiation governance action    Then I should be able to see: - The new major protocol version |  |  |
| The governance action as displayed should include a link to its metadata | Given that I am reviewing the details of a specific governance action  When I click on the “view other details” link on the governance action details page  Then I will be shown a warning asking if I want to continue to an external url (which will be displayed). If I continue then the external url will open in a new tab. |  |  | Voltaire | N |
| Verify the integrity of a governance action’s metadata | Given that I am looking at a governance action    Then I can see whether a hash of that governance action’s metadata matches the *metadata hash* included in the anchor. |  |  | Voltaire | N |
| CH.VO8 | As a potential Constitutional Committee Member **I want to become a Constitutional Committee Member** so that I can vote on the corresponding governance actions | Create a set of keys | Given that I have the CLI open in front of me, when I run the corresponding command, then I can verify that I have created a new key pair |  |  | Voltaire | N |
| Include these keys in a New Committee Cold Key Hash or Script Hash | Given that I am using CLI and I have created my set of keys, when they are registered in the ledger, then I can verify that my set of keys corresponds to a CC member |  |  | Voltaire | N |
| Create an authorization certificate | Given that I am using CLI and I have created my set of cold/hot key pairs, when I run the corresponding command, then I can verify that the certificate is stored on-chain and it delegates rights from the cold key to the hot key |  |  | CIP-1694  [**L993**](https://github.com/cardano-foundation/CIPs/blob/de76120a30552ef620d96401ed7fd3e5bca1f062/CIP-1694/README.md?plain=1#L993) | N |
| Only votes from active Committee members are considered. | Given that my set of keys does not correspond to an active CC member, when I try to vote as a CC member using CLI, then I get an error message telling me that my credentials are not valid |  |  | CIP-1694  [**L993**](https://github.com/cardano-foundation/CIPs/blob/de76120a30552ef620d96401ed7fd3e5bca1f062/CIP-1694/README.md?plain=1#L993) | N |
| CH.VO9 | **As a CC Member I want to review a governance action** so that I can scrutinise whether or not it is aligned with the Constitution | Access GovTool | Given that I am on the GovTool homepage, when I click the “Governance actions” in the topbar then I am sent to the governance actions page. |  |  | Voltaire | N |
| See list of current governance actions | Given that I am on the Governance Actions page When I review the governance actions available to view on the page, then all of the non expired/ ratified/enacted governance actions. |  |  | Voltaire | N |
| See details of a specific governance action | Given that I am on the Governance Actions page, When I click on an individual governance action, Then I can see the relevant information |  |  | Voltaire | N |
| See metadata | Given that I am looking at an individual governance action page, when I click on “view other details” and I click on continue in the warning message, Then an external url will open in a new tab. |  |  | Voltaire | N |
| Verify metadata integrity | Given that I am looking at a governance action then I can see whether a hash of that governance action’s metadata matches the metadata hash included in the anchor. |  |  | Voltaire | N |
| CH.VO10 | **As a CC Member I want to vote on a governance action** so that I can approve governance actions that align with the constitution | Find a specific governance action using CLI | Given that I am using CLI and I know the governance action ID, When I run the command to review a specific governance action, Then I can see the details of that individual governance action. |  |  | Voltaire | N |
| Vote as a Constitutional Committee member using CLI | Given that I am using CLI and I know the governance action ID, When I build the transaction to vote on a governance action, Then I can specify my role (ccm), the action ID, and the intent to vote (YES/NO/ABSTAIN) |  |  | Voltaire | N |
| Given that I am using CLI and I have built the transaction to vote on a specific governance action, When I run the command to sign the transaction, Then I can submit the transaction and pay the transaction costs |  |  | Voltaire | N |
| Provide rationale for vote with a metadata anchor | Given that I am using CLI, When I am building a transaction to vote on a specific governance action, Then I can use the metadata anchor to provide a rationale for my vote |  |  | Voltaire | N |
| Change vote | Given that I have submitted a vote on a specific governance action, When I vote again on the same governance action, Then I can change the intent to vote (YES/NO/ABSTAIN) |  |  | Voltaire | N |
| CH.VO11 | **As a CC Member I want to change my voting credentials** so I can manage my organisation’s voting | Create a new authorization certificate | Given that I am a CC member and I have enough funds to pay the transaction fees, When I create a new authorisation certificate using the same cold key, Then a new hot key is authorised to vote as a valid CC member |  |  | Voltaire | N |
| CH.VO12 | **As a CC Member I want to retire** so my credentials are no longer valid | Create a resignation certificate | Given that I am a CC member and I have enough funds to pay the transaction fees, When I create a resignation certificate, Then my cold key and the derived hote keys are no longer valid to vote as a CC member |  |  | Voltaire | N |
| CH.CLI1 | **As a Plutus developer I want the Node to support Plutus V3 requirements for CIP-1694** | *Correct operation of BLS primitives.*  *Correct operation of new bitwise primitives.*  *New Plutus V3 cost model is used correctly for the new primitives.*  *Plutus scripts are correctly used for DRep voting.*  *Plutus scripts on the constitution correctly control CC voting.*  *Benchmark execution times are within acceptable range.* | Detail pending |  | Detail pending | API/CLI |  |
| CH.API11 | **Ledger API User Story** | This is a placeholder | Detail pending |  |  | API/CLI | UPDATE/REMOVE |
| VO16 | **SPO Interface User Story** |  |  |  |  | API/CLI | UPDATE/REMOVE |
| INT1 | As a user of the system I want governance actions to be correctly submitted on chain |  | Detail pending |  |  |  |  |
| *INT2* | *As a user of the system I want All actions to be ratified on-chain correctly according to the CIP-1694 rules* | *(voter type, thresholds, DRep activity)* | Detail pending |  |  |  |  |
| *INT3* | *As a user of the system I want All governance actions to be enacted on time on-chain following ratification* | **Timely** |  |  |  |  |  |
| **Correctly according to CIP-1694 rules:***Parameter updates, Info, Hard Forks, No Confidence, New Committee, New Constitution, Treasury Withdrawals. This will expand into many requirements.* |  |  |  |  |  |
| **SYSTEM LEVEL (not fully completed yet)** | | | | | | | |
| [CIP1](https://docs.google.com/document/d/1Ymj_Foh3lVYq2asI5WEdQYBud5B8KbJcw4jpUwuv5p8/edit#bookmark=id.yzo1aziuwsxs) | **As** any persona  **I want** thehash value of the off-chain Constitution to be recorded on-chain  **So that** I can verify the authenticity of the off-chain document | Hash value of the off-chain Constitution is recorded on-chain | Given that I access a Cardano Explorer and that it display the ledger states accurately, Then I can access a map of the current constitution hash and the URL of the off-chain document |  | **Detail Pending** | CIP-1694  [**L958**](https://github.com/cardano-foundation/CIPs/blob/de76120a30552ef620d96401ed7fd3e5bca1f062/CIP-1694/README.md?plain=1#L958C19-L958C120) |  |
| [CIP2](https://docs.google.com/document/d/1Ymj_Foh3lVYq2asI5WEdQYBud5B8KbJcw4jpUwuv5p8/edit#bookmark=id.elqu6qa8ltse) | **As** an ada holder  **I want** thekey hash of active and expired Committee Members and their terms to be registered on-chain  **So that** the system can count their votes | Committee members key hashes must be recorded on chain and be publicly accessible  ADA holder must be able to check that the key hashes of active and expired committee members are registered on-chain status | Given that the node is up to date, then it should include a record for each committee member detailing their key-hashes, term information (end epochs), and the corresponding cold to hot key authorization maps. |  | **Detail Pending** | CIP-1694  [**L993**](https://github.com/cardano-foundation/CIPs/blob/de76120a30552ef620d96401ed7fd3e5bca1f062/CIP-1694/README.md?plain=1#L993) |  |
| [CIP3](https://docs.google.com/document/d/1Ymj_Foh3lVYq2asI5WEdQYBud5B8KbJcw4jpUwuv5p8/edit#bookmark=id.pddqksyfshgc) | **As an** ada holder  **I want** to be able to submit a proposal to replace all or part of the current constitutional committee  **So that** committee members that have lost confidence of ada holders can be removed from their duties. | It must be possible to replace the constitutional committee via a governance action | Any ada holder can submit a governance action that proposes to add or remove one or many new committee members.    Given that a governance action to update the committee has been proposed And both SPO and DRep YES votes are equal or greater than the required threshold (depending on state of no-confidence or normal state)  The governance action is ratified and enacted automatically at the next epoch transition, otherwise it’s expired. |  | **Detail Pending** | CIP-1694  [**L997**](https://github.com/cardano-foundation/CIPs/blob/de76120a30552ef620d96401ed7fd3e5bca1f062/CIP-1694/README.md?plain=1#L997C1-L997C78) |  |
| [CIP4](https://docs.google.com/document/d/1Ymj_Foh3lVYq2asI5WEdQYBud5B8KbJcw4jpUwuv5p8/edit#bookmark=id.mca6cvcoht53) | **As an** ada holder  I want the size of the constitutional committee  To be not-fixed  So that I can propose a different size via a governance action | Size of the constitutional committee must be variable | **Detail Pending** |  | **Detail Pending** | [**L1009**](https://github.com/cardano-foundation/CIPs/blob/ada2cf6a9e27fbdf37ca1be0780371fbab2b7748/CIP-1694/README.md?plain=1#L1009) |  |
| [CIP5](https://docs.google.com/document/d/1Ymj_Foh3lVYq2asI5WEdQYBud5B8KbJcw4jpUwuv5p8/edit#bookmark=id.tdyxk950f5v8) | As an ada holder,  I want that the committee threshold (the fraction of committee is not fixed  So that I can propose a different threshold via a governance action | **Size of** committee threshold must be variable | **Detail Pending** |  | **Detail Pending** | [**L1011**](https://github.com/cardano-foundation/CIPs/blob/ada2cf6a9e27fbdf37ca1be0780371fbab2b7748/CIP-1694/README.md?plain=1#L1011) |  |
| CIP6 | As an ada holder  I want to be able to have elect an empty committee if the community wishes to abolish the constitutional committee entirely  So that SPO and Dreps can continue to ratify governance actions without the need of a constitutional committee | Users must be able to elect an empty committee    Ratifying governance actions must be possible without constitutional committee | **Detail Pending** |  | **Detail Pending** | [**L1014**](https://github.com/cardano-foundation/CIPs/blob/ada2cf6a9e27fbdf37ca1be0780371fbab2b7748/CIP-1694/README.md?plain=1#L1014) |  |
| CIP7 | **As** an Ada holder,  **I want** to submit a governance action,  **So that** it can be considered by the governance bodies/process. | **There must be a command in in the Cardano CLI allowing the user to submit a governance action**    The input must include validation checks for format and criteria    After successful validation the  user must be informed of confirmation through the CLI    An action should have an accepted or rejected status? | **Detail Pending** |  | **Detail Pending** | CIP-1694 |  |
| CIP8 | **As** a delegate representative (DRep),  **I want** to vote on submitted governance actions,  **So that** I can represent the interests of my delegates. | DReps must have access to the list of current governance actions (can we specify where they get it?).  DReps should be able to cast their vote on each action (through the CL or through an app)?  An action should have an accepted or rejected status? | Given that a Drep has casted a vote the system accurately records and reflects these votes. |  | **Detail Pending** | CIP-1694 |  |
| CIP9 | **As** an Ada holder,  **I want** to view the status of governance actions,  **So that** I am informed about the governance process. | Users must be able to view a list of governance actions with their current status.  The status must include stages like submission, voting, ratification, timeline.  The system must provide real-time updates on the status. | **Detail Pending** |  | **Detail Pending** | CIP-1694 |  |
| CIP10 | **As** an Ada holder,  **I want** to delegate my voting rights to a DRep,  **So that** my interests are represented in the governance process. | ADA holders can select a DRep to delegate their voting rights.  The delegation process should be user-friendly and must be secure.  The system must confirms successful delegation. | **Detail Pending** |  |  |  |  |
| CIP11 | **As** an SPO,  **I want** to participate in the governance voting,  **So that** I can contribute to the decision-making process. | SPOs must have access to governance actions for voting.  SPOs can cast their votes based on the ADA staked in their pools.  The system accurately reflects the votes of SPOs. | **Detail Pending** |  |  |  |  |
| **CARDANO CLI US** | | | | | | | |
| CLI01 | **As** an Ada holder,  **I want** to obtain the hash of the off-chain text of a Constitution  **So that** I can compare it against the hash registered on-chain to verify its authenticity. | When I provide the off-chain text of the Constitution, the cardano-cli should calculate and return the corresponding blake2b-256 hash of the document. | Given that a holder provides the offchain text of the constitution then cardo-cli should return the corresponding blake2b-256 hash  Given that it is the same document, the resulting hash should match the one registered on-chain. |  |  | [L958](https://github.com/cardano-foundation/CIPs/blob/d6c5ad3a77b4684bc19f3cafb75b4886a67c9a31/CIP-1694/README.md?plain=1#L958) |  |
| CLI02 | **As** an Ada holder,  **I want** to generate the hash of the off-chain text for a proposed Constitution  **So that** the hash can be utilized in a governance action. | When I provide the off-chain text of the Constitution, the cardano-cli should calculate and return the corresponding blake2b-256 hash of the document. | Given that a holder provides the offchain text of the constitution then cardo-cli should return the corresponding blake2b-256 hash  Given that it is the same document, the resulting hash should match the one registered on-chain. |  |  | [L958](https://github.com/cardano-foundation/CIPs/blob/d6c5ad3a77b4684bc19f3cafb75b4886a67c9a31/CIP-1694/README.md?plain=1#L958) |  |
| CLI03 | **As** a potential Constitutional Committee member,  **I want** to generate COLD key pair  **So that** I can be proposed for the Committee in a Governance action | The feature implementation should include a new command:   cardano-cli conway governance committee key-gen-cold | Typingcardano-cli conway governance committee key-gen-cold with accepted input parameters generates a COLD key pair. If a parameter or the command format is incorrect an error is raised |  |  | [L966](https://github.com/cardano-foundation/CIPs/blob/d6c5ad3a77b4684bc19f3cafb75b4886a67c9a31/CIP-1694/README.md?plain=1#L966C33-L966C52)  [L993](https://github.com/cardano-foundation/CIPs/blob/d6c5ad3a77b4684bc19f3cafb75b4886a67c9a31/CIP-1694/README.md?plain=1#L993) |  |
| Includes a corresponding CLI usage describing the feature, how to use it and the types of the inputs and outputs. | Typingcardano-cli conway governance committee key-gen-cold —-help displays command usage |  |  | [L966](https://github.com/cardano-foundation/CIPs/blob/d6c5ad3a77b4684bc19f3cafb75b4886a67c9a31/CIP-1694/README.md?plain=1#L966C33-L966C52)  [L993](https://github.com/cardano-foundation/CIPs/blob/d6c5ad3a77b4684bc19f3cafb75b4886a67c9a31/CIP-1694/README.md?plain=1#L993) |  |
| The command must accept two flags  --cold-verification-key-file  --cold-signing-key-file | Both flags are mandatory and each produces the corresponding verification or signing key file |  |  | [L966](https://github.com/cardano-foundation/CIPs/blob/d6c5ad3a77b4684bc19f3cafb75b4886a67c9a31/CIP-1694/README.md?plain=1#L966C33-L966C52)  [L993](https://github.com/cardano-foundation/CIPs/blob/d6c5ad3a77b4684bc19f3cafb75b4886a67c9a31/CIP-1694/README.md?plain=1#L993) |  |
| The generated key pair should be stored in the specified files:    the verification key is saved in the file specified by --cold-verification-key-file    the signing key saved in the file specified by  --cold-signing-key-file | Given that the user specifies a valid path and file name, Then the keys are saved on that file and location. |  |  | [L966](https://github.com/cardano-foundation/CIPs/blob/d6c5ad3a77b4684bc19f3cafb75b4886a67c9a31/CIP-1694/README.md?plain=1#L966C33-L966C52)  [L993](https://github.com/cardano-foundation/CIPs/blob/d6c5ad3a77b4684bc19f3cafb75b4886a67c9a31/CIP-1694/README.md?plain=1#L993) |  |
| The generated keys should adhere to text envelope format used for other credentials | Given that the cli has created the verification and signing keys, then these conform to the text envelope format used consisting of a json object with type, description and cborhex fields. |  |  | [L966](https://github.com/cardano-foundation/CIPs/blob/d6c5ad3a77b4684bc19f3cafb75b4886a67c9a31/CIP-1694/README.md?plain=1#L966C33-L966C52)  [L993](https://github.com/cardano-foundation/CIPs/blob/d6c5ad3a77b4684bc19f3cafb75b4886a67c9a31/CIP-1694/README.md?plain=1#L993) |  |
| The signing key text envelope contains the correct type, description, and CBOR value.    **Type** "ConstitutionalCommitteeColdSigningKey\_ed25519"    **Description**  "Constitutional Committee Cold Signing Key" | Given that the signing key is saved on a text envelope format, the type and description fields are:  **Type** "ConstitutionalCommitteeColdSigningKey\_ed25519"    **Description**  "Constitutional Committee Cold Signing Key" |  |  | [L966](https://github.com/cardano-foundation/CIPs/blob/d6c5ad3a77b4684bc19f3cafb75b4886a67c9a31/CIP-1694/README.md?plain=1#L966C33-L966C52)  [L993](https://github.com/cardano-foundation/CIPs/blob/d6c5ad3a77b4684bc19f3cafb75b4886a67c9a31/CIP-1694/README.md?plain=1#L993) |  |
| The verification key text envelope has:  **Type** "CConstitutionalCommitteeColdVerificationKey\_ed25519"  **Description**  "Constitutional Committee Cold Verification Key" | Given that the verification key is saved on a text envelope format, the type and description fields are:    **Type** "CConstitutionalCommitteeColdVerificationKey\_ed25519"  **Description**  "Constitutional Committee Cold Verification Key" |  |  | [L966](https://github.com/cardano-foundation/CIPs/blob/d6c5ad3a77b4684bc19f3cafb75b4886a67c9a31/CIP-1694/README.md?plain=1#L966C33-L966C52)  [L993](https://github.com/cardano-foundation/CIPs/blob/d6c5ad3a77b4684bc19f3cafb75b4886a67c9a31/CIP-1694/README.md?plain=1#L993) |  |
| Failing to provide a file name for any of    --cold-verification-key-file  --cold-signing-key-file    fails with an appropriate error message. | **Given** the user has not inputed either --cold-verification-key-file OR  --cold-signing-key-file parameters **THEN** the command fails and returns an error to the users informing them to fill those parameters in. The error message should prompt the user to consult the command usage (--help) |  |  | [L966](https://github.com/cardano-foundation/CIPs/blob/d6c5ad3a77b4684bc19f3cafb75b4886a67c9a31/CIP-1694/README.md?plain=1#L966C33-L966C52)  [L993](https://github.com/cardano-foundation/CIPs/blob/d6c5ad3a77b4684bc19f3cafb75b4886a67c9a31/CIP-1694/README.md?plain=1#L993) |  |
| CLI04 | **As** potential Constitutional Committee member,  **I want** to generate HOT key pair  **So that** I can authorise the Hot key to sign votes on behalf of the Cold key | The feature implementation should include a new command:   cardano-cli conway governance committee key-gen-hot | Given that a potential constitutional committee member has entered the cardano-cli conway governance committee key-gen-hot command with all the required and correct parameters then the command is executed successfully and a HOT key pair is generated.  If a parameter or the command format is incorrect an error is raised |  |  | [L966](https://github.com/cardano-foundation/CIPs/blob/d6c5ad3a77b4684bc19f3cafb75b4886a67c9a31/CIP-1694/README.md?plain=1#L966C33-L966C52)  [L993](https://github.com/cardano-foundation/CIPs/blob/d6c5ad3a77b4684bc19f3cafb75b4886a67c9a31/CIP-1694/README.md?plain=1#L993) |  |
| Includes a corresponding CLI usage describing the feature, how to use it and the types of the inputs and outputs. | Typingcardano-cli conway governance committee key-gen-hot displays command usage |  | **Detail Pending** | [L966](https://github.com/cardano-foundation/CIPs/blob/d6c5ad3a77b4684bc19f3cafb75b4886a67c9a31/CIP-1694/README.md?plain=1#L966C33-L966C52)  [L993](https://github.com/cardano-foundation/CIPs/blob/d6c5ad3a77b4684bc19f3cafb75b4886a67c9a31/CIP-1694/README.md?plain=1#L993) |  |
| The command must accept two flags  --verification-key-file  --signing-key-file | Both flags are mandatory and each produces the corresponding verification or signing key file |  | **Detail Pending** | [L966](https://github.com/cardano-foundation/CIPs/blob/d6c5ad3a77b4684bc19f3cafb75b4886a67c9a31/CIP-1694/README.md?plain=1#L966C33-L966C52)  [L993](https://github.com/cardano-foundation/CIPs/blob/d6c5ad3a77b4684bc19f3cafb75b4886a67c9a31/CIP-1694/README.md?plain=1#L993) |  |
| The generated key pair should be stored in the specified files:    the verification key saved in the file specified by  --verification-key-file    the signing key saved in the file specified by  --signing-key-file | Given that the user specifies a valid path and file name, Then the keys are saved on that file and location. |  |  | [L966](https://github.com/cardano-foundation/CIPs/blob/d6c5ad3a77b4684bc19f3cafb75b4886a67c9a31/CIP-1694/README.md?plain=1#L966C33-L966C52)  [L993](https://github.com/cardano-foundation/CIPs/blob/d6c5ad3a77b4684bc19f3cafb75b4886a67c9a31/CIP-1694/README.md?plain=1#L993) |  |
| The generated keys should adhere to text envelope format used for other credentials | Given that the cli has created the verification and signing keys, then these conform to the text envelope format used consisting of a json object with type, description and cbor hex fields. |  |  | [L966](https://github.com/cardano-foundation/CIPs/blob/d6c5ad3a77b4684bc19f3cafb75b4886a67c9a31/CIP-1694/README.md?plain=1#L966C33-L966C52)  [L993](https://github.com/cardano-foundation/CIPs/blob/d6c5ad3a77b4684bc19f3cafb75b4886a67c9a31/CIP-1694/README.md?plain=1#L993) |  |
| The **signing** key text envelope contains the correct type, description, and CBOR value.    **Type** "ConstitutionalCommitteeColdSigningKey\_ed25519"    **Description**  "Constitutional Committee Cold Signing Key" | Given that the signing key is saved on a text envelope format, the type and description fields are:  **Type** "ConstitutionalCommitteeColdSigningKey\_ed25519"    **Description**  "Constitutional Committee Cold Signing Key" |  |  | [L966](https://github.com/cardano-foundation/CIPs/blob/d6c5ad3a77b4684bc19f3cafb75b4886a67c9a31/CIP-1694/README.md?plain=1#L966C33-L966C52)  [L993](https://github.com/cardano-foundation/CIPs/blob/d6c5ad3a77b4684bc19f3cafb75b4886a67c9a31/CIP-1694/README.md?plain=1#L993) |  |
| The **verification** key text envelope has:  **Type** "CConstitutionalCommitteeColdVerificationKey\_ed25519"  **Description** "Constitutional Committee Cold Verification Key" | Given that the verification key is saved on a text envelope format, the type and description fields are:  **Type** "CConstitutionalCommitteeColdVerificationKey\_ed25519"  **Description** "Constitutional Committee Cold Verification Key" |  |  | [L966](https://github.com/cardano-foundation/CIPs/blob/d6c5ad3a77b4684bc19f3cafb75b4886a67c9a31/CIP-1694/README.md?plain=1#L966C33-L966C52)  [L993](https://github.com/cardano-foundation/CIPs/blob/d6c5ad3a77b4684bc19f3cafb75b4886a67c9a31/CIP-1694/README.md?plain=1#L993) |  |
| Failing to provide a file name for any of the flags    --verification-key-file  --signing-key-file    fails with an appropriate error message. | **Given** the user has not inputted either --verification-key-file OR  --signing-key-file parameters **THEN** the command fails and returns an error to the users informing them to fill those parameters in. The error message should prompt the user to consult the command usage (--help) |  |  | [L966](https://github.com/cardano-foundation/CIPs/blob/d6c5ad3a77b4684bc19f3cafb75b4886a67c9a31/CIP-1694/README.md?plain=1#L966C33-L966C52)  [L993](https://github.com/cardano-foundation/CIPs/blob/d6c5ad3a77b4684bc19f3cafb75b4886a67c9a31/CIP-1694/README.md?plain=1#L993) |  |
| CLI05 | As a committee member  I want to issue a authorization certificate from my cold key to a hot key    So that I can sign my votes using the hot key and keep the cold key in cold storage and can authorise a new hot key in case the original one is compromised. | The feature implementation should include a new command in the cardano-cli:    cardano-cli conway governance committee create-hot-key-authorization-certificate | Typingcardano-cli conway governance committee create-hot-key-authorization-certificate with accepted input parameters generates a hot key authorization certificate. If a parameter or the command format is incorrect an error is raised |  |  | [L993](https://github.com/cardano-foundation/CIPs/blob/0f64c52fe1350b4e93c23f6bced92d4ca576538e/CIP-1694/README.md?plain=1#L993) |  |
| The command should accept the necessary flags:    --cold-verification-key, --cold-verification-key-file, --cold-verification-key-hash, --hot-verification-key, --hot-verification-key-file, --hot-verification-key-hash, and --out-file. | The command allows passing credentials as follows  Cold verification key <- string  Cold verification key file <- file  Cold verification key hash <- string  Hot verification key <- string  Hot verification key file <- file  Hot verification key hash <- string   Failing to provide the right input results in a clear error message that helps the user to identify the problem |  |  | [L993](https://github.com/cardano-foundation/CIPs/blob/0f64c52fe1350b4e93c23f6bced92d4ca576538e/CIP-1694/README.md?plain=1#L993) |  |
| Running the command with the appropriate flags should generate a hot key authorization certificate and be saved in the specified output file. | Given that the user specifies a valid path and file name, then the command produces a Cold to Hot authorization certificate on the right location and name. |  |  | [L993](https://github.com/cardano-foundation/CIPs/blob/0f64c52fe1350b4e93c23f6bced92d4ca576538e/CIP-1694/README.md?plain=1#L993) |  |
| The hot key authorization certificate should follow the text envelope format of other existing certificates, including the type, description, and CBOR hex value.    {  "type": "CertificateConway",  "description": "Constitutional Committee Hot Key Authorization Certificate",  "cborHex": ""  } | Given that the authorization certificate is saved, then it is in a text envelope format consisting of a json object with type, description and cbor hex fields. |  |  | [L993](https://github.com/cardano-foundation/CIPs/blob/0f64c52fe1350b4e93c23f6bced92d4ca576538e/CIP-1694/README.md?plain=1#L993) |  |
| The type of the certificate should be: "CertificateConway" | The type field ont the text envelope of the certificate is  "CertificateConway" |  |  | [L993](https://github.com/cardano-foundation/CIPs/blob/0f64c52fe1350b4e93c23f6bced92d4ca576538e/CIP-1694/README.md?plain=1#L993) |  |
| The description of the certificate should be:    "Committee HotKey Authorization Certificate" | The description field on the text envelope of the certificate is  "Committee HotKey Authorization Certificate" |  |  | [L993](https://github.com/cardano-foundation/CIPs/blob/0f64c52fe1350b4e93c23f6bced92d4ca576538e/CIP-1694/README.md?plain=1#L993) |  |
| The certificate must comply with the cddl:    auth\_committee\_hot\_cert = (14, committee\_cold\_credential, committee\_hot\_credential) | Given that the authorization certificate is saved,then it is on the form of a text envelope consisting of a json object with type, description and cbor hex fields , where  Cbor hex conforms to the authorization certificate as defined on the cddl  auth\_committee\_hot\_cert = (14, committee\_cold\_credential, committee\_hot\_credential) |  |  | [L993](https://github.com/cardano-foundation/CIPs/blob/0f64c52fe1350b4e93c23f6bced92d4ca576538e/CIP-1694/README.md?plain=1#L993) |  |
| The command should handle potential errors, such as missing or invalid flags, and provide appropriate error messages indicating the missing or required parameters. | The command should handle potential errors, such as missing or invalid flags, and provide appropriate error messages indicating the missing or required parameters. |  |  | [L993](https://github.com/cardano-foundation/CIPs/blob/0f64c52fe1350b4e93c23f6bced92d4ca576538e/CIP-1694/README.md?plain=1#L993) |  |
| Documentation should be provided, including a corresponding CLI usage, describing the feature, its purpose, and how to use it, along with the expected types of inputs and outputs. | Typingcardano-cli conway governance committee create-hot-key-authorization-certificate —-help displays command usage |  |  | [L993](https://github.com/cardano-foundation/CIPs/blob/0f64c52fe1350b4e93c23f6bced92d4ca576538e/CIP-1694/README.md?plain=1#L993) |  |
| CLI06 | As a potential constitutional committee member  I want to generate the key hashes for my cold and hot keys  So that they can be used by third parties to propose me as a new committee member and for identification purposes once Ive been elected as committee member | The command generates the blake2b 256 hash of the verification key  cardano-cli conway governance committee key-hash | Typingcardano-cli conway governance committee key-hash with accepted input parameters generates blake2b 256 hash of the verification key. If a parameter or the command format is incorrect an error is raised |  |  |  |  |
| Accepts the flags  --verification-key --verification-key-file  To provide the verification key. | Both flags are mandatory and each produces the corresponding verification or signing key file |  |  |  |  |
| The command gives the option –out-file. If the --out-file flag is provided, the key hash should be saved to the specified file. | Given that the user specifies the –out-file parameter and a valid path and file name, Then the key hash is saved on that file and location. |  |  |  |  |
| If the --out-file flag is not provided, the key hash should be printed to the standard output (stdout) in a readable format. | Given that the user has not specified the –out-file parameter then the key hash must returned in clear text through stdout (cmd output) |  |  |  |  |
| The command should handle potential errors, such as missing flags or invalid input, and provide appropriate error messages or exceptions to guide the user. | Given that the user has executed the correct command but has either filled incorrectly any of the parameters, missed any mandatory flag and/or parameter then an exception should be raised and an error message should be returned with clear indication as to how to fix the issue. When not feasible, the user should be directed to the usage page of the command (cardano-cli conway governance committee key-hash --help) |  |  |  |  |
| Documentation should be provided, including a corresponding CLI usage, describing the feature, its purpose, and how to use it, along with the expected types of inputs and outputs. | Typing cardano-cli conway governance committee key-hash --help display the command usage page |  |  |  |  |
| CLI07 | As a constitutional committee member  I want to be able to generate a resignation certificate  So that i can submit it it to the chain on a transaction to signal to the ada holders that I’m resigning from my duties as cc member | The command should be implemented in the cardano-cli as cardano-cli governance committee create-cold-resignation-certificate | Typing cardano-cli governance committee create-cold-resignation-certificate with accepted input parameters generates a cold resignation certificate. |  |  |  |  |
| The command should accept the following ways to supply credentials: cold verification key file, the cold verification key or the cold verification key hash so it accepts the following flags:    --cold-verification-key,  --cold-verification-key-file,  --cold-verification-key-hash. | The command allows passing credentials as follows    Cold verification key <- string  Cold verification key file <- file  Cold verification key hash <- string    If any of the inputs has the wrong format the cli raises an arrow message indicating the missing or incorrect parameter. |  |  |  |  |
| The command allows an optional anchor (url/hash) for the resigning  cc member to express their motivation for resigning. | The command allows for an OPTIONAL anchor comprised of a URL containing a document where the resigning CC member expresses the motivations and the hash of the document  The CLI does not check for the validity of the URL, the contents, or that the documents and declared hash match. |  |  |  |  |
| The resignation certificate should be saved in the specified output file using the --out-file flag. | Given that the CC member issuing the resignation certificate provides a valid path and file name, then the certificate is saved on the specified location. |  |  |  |  |
| The generated resignation certificate should be consistent with the text envelope format of existing certificates in Cardano, including the specified JSON structure with the correct type, description, and CBOR-encoded value.        {  "type": "CertificateConway",  "description": "Constitutional Committee Cold Key Resignation Certificate",  "cborHex": ""  } | Given that the resignation certificate is saved, then it is in a text envelope format consisting of a json object with type, description and cbor hex fields. |  |  |  |  |
| The certificate type should be ConwayResignCommitteeColdKey to indicate a resignation of the Committee Cold Key. | Given that the certificate is in a text envelope format, then its type is   Type: ConwayResignCommitteeColdKey |  |  |  |  |
| The certificate description should be    "description": "Constitutional Committee Cold Key Resignation Certificate" | Given that the certificate is in a text envelope format, then its description field is   "description": "Constitutional Committee Cold Key Resignation Certificate" |  |  |  |  |
| The command should generate a resignation certificate according to the conway cddl:  **resign\_committee\_cold\_cert = (15, committee\_cold\_credential, anchor / null)** | Given that the certificate is in a text envelope format, then its cbor hex field conforms to the conway cddl  **resign\_committee\_cold\_cert = (15, committee\_cold\_credential, anchor / null)** |  |  |  |  |
| The command should handle potential errors, such as missing or invalid flags or keys, and provide appropriate error messages indicating the missing or required parameters. | If any required input parameter is missing or incorrect, the command should raise an error indicating the missing or incorrect parameter. |  |  |  |  |
| Documentation should be provided, including a corresponding CLI usage, describing the feature, its purpose, and how to use it, along with the expected types of inputs and outputs. | Given that the user has executed the correct command but has either filled incorrectly any of the parameters, missed any mandatory flag and/or parameter then an exception should be raised and an error message should be returned with clear indication as to how to fix the issue. When not feasible, the user should be directed to the usage page of the command (cardano-cli governance committee create-cold-resignation-certificate --help) |  |  |  |  |
| CLI08 | As an ada holder  I want to generate Ed25519 keys  So that I can register as a DRep | The command is implemented as   cardano-cli conway governance drep key-gen | Typing cardano-cli conway governance drep key-gen with accepted input parameters generates an Ed25519 key pair |  |  |  |  |
| The command supports the --verification-key-file FILE option to specify the file where the generated verification key will be saved. | The flag--verification-key-file is mandatory, |  |  |  |  |
| The command supports the --signing-key-file FILE option to specify the file where the generated signing key will be saved. File is saved on the specified path | The flag--signing-key-file is mandatory, |  |  |  |  |
| When the command is executed, it generates both the verification key and signing key File is saved on the specified path | Given that the user specifies valid paths and file names then, the corresponding verification and signing key files are saved on the paths specified by the user.    If the paths do not exist or are inaccessible, an error message is raised. |  |  |  |  |
| The key generation should follow the current design for key generation and output in an envelope format. | Given that the verification and signing key files are saved, then they are in a text envelope format consisting of a json object with type, description and cbor hex fields, where:    {  "type": "DRepVerificationKey\_ed25519",  "description": "Delegate Representative Verification Key",  "cborHex": ""  }    {  "type": "DRepSigningKey\_ed25519",  "description": "Delegate Representative VSigning Key",  "cborHex": ""  } |  |  |  |  |
| CLI09 | As a DRep  I want to generate a Drep Id  So that ada holder can use it to delegate their votes to me and my voting record can be tracked | The command is implemented as cardano-cli conway governance drep id  and generates the blake2b-224 hash digest of the serialised DRep credential (verification key) | Typing cardano-cli conway governance drep id  with accepted input parameters generates a DRep ID, the blake2b-224 hash digest of the serialised DRep credential (verification key) |  |  |  |  |
| The command accepts supplying the verification key with either  --drep-verification-key  –drep-verification-key-file options | Using one of the flags is mandatory, the flags are mutually exclusive and take a string and file respectively:  --drep-verification-key <- string  –drep-verification-key-file <- file |  |  |  |  |
| The command supports the --out-file FILE option (optional) to enable users to save the generated DRep ID to the specified file. | Given that the user wants to save the DrepId to a file, and specifies a valid path and file name, then, using the –out-file flag saves the drep id to a file on the specified location.  If the specified path does not exist or is inaccessible the cli returns an error message.   If the –out-flag is not used the Drep ID is printed on stdout |  |  |  |  |
| The command should support the an option to specify the output format  Accepted output formats are "hex" and "bech32", where "bech32" is the default format. | Given that the optional --output-format flag is used, and “bech32” is given as an argument, then the DRep Id is printed in bech32 format with the “drep” prefix.    Given that the optional --output-format flag is used, and the hex is given as an argument, then the DRep Id is printed in hex format    If the --output-format STRING option is **not** specified, the DRep ID should be displayed in "bech32" format by default. |  |  |  |  |
| The command supports an optional –out-file to save the Drep Id to a file | If the --out-file FILE option is specified, the generated DRep ID should be saved to the specified file, otherwise it is printed on stdout |  |  |  |  |
| The feature is well-documented, providing clear usage instructions for the cardano-cli conway governance drep id command. | Typing cardano-cli conway governance drep id --help  displays the command usage page |  |  |  |  |
| The command handles errors gracefully and provides helpful error messages when required options are missing or invalid inputs are provided. | If any required input parameter is missing or incorrect, the command should raise an error indicating the missing or incorrect parameter. |  |  |  |  |
| CLI10 | As a DRep  I want to generate a Drep registration certificate  So that I can submit it on a transaction and the ada holders can delegate their votes to me. | The command is implemented as  cardano-cli conway governance drep registration-certificate | Running  cardano-cli conway governance drep registration-certificate with accepted input parameters generates a drep registration certificate. |  |  |  |  |
| The command requires the user to provide the amount required for drep key deposit to comply with the conway cddl -> “reg\_drep\_cert” | The flag --key-reg-deposit-amt is mandatory and takes the deposit amount in lovelace as argument. |  |  |  |  |
| The command allows the user to provide the DRep credential in the following ways:  Drep verification key  Drep verification key file  Drep Id | The command presents the options to pass the Drep credential:    --drep-verification-key STRING  --drep-verification-key-file FILE  --drep-id STRING    Using one of these is mandatory but are mutually exclusive.    Failing to provide the right input results in a clear error message that helps the user to identify the problem |  |  |  |  |
| The command allows adding an optional anchor (url/hash) to submit any drep metadata. | Given that the user wants to include an anchor with the Dreps metadata, then the command requires both the url and the hash to be provided. Only url or only hash is not allowed.  --drep-metadata-url TEXT --drep-metadata-hash HASH  Failing to provide the right input results in a clear error message that helps the user to identify the problem |  |  |  |  |
| When the --drep-metadata-url and --drep-metadata-hash    are provided, the resulting certificate should include the url and hash on the anchor position, otherwise the field is null. | The anchor is included in the certificate on the last position:  reg\_drep\_cert = (16, drep\_credential, coin, anchor / null) |  |  |  |  |
| The --out-file FILE option is mandatory and used to specify the file where the generated DRep registration certificate will be saved. | The flag --out-file is mandatory and takes the file path and name as an argument.  Failure to provide the flag and its argument generates an exception. |  |  |  |  |
| The command generates a registration certificate compliant with the conway cddl | reg\_drep\_cert = (16, drep\_credential, coin, anchor / null) |  |  |  |  |
| The certificate should be on a text envelope format, similar to what we have for stake pools registration certificates | Given that the certificate is saved, then it is in a text envelope format consisting of a json object with type, description and cbor hex fields, where:  {  "type": "CertificateConway",  "description": "DRep Registration Certificate",  "cborHex": "00000000"  }   |  |  |  |  |
| The feature implementation should be well-documented, providing clear usage instructions for the cardano-cli conway governance drep registration-certificate command. | **Detail Pending** |  |  |  |  |
| The command should handle errors gracefully and provide helpful error messages when required options are missing or invalid inputs are provided. | **Detail Pending** |  |  |  |  |
| CLI11 | As a DRep  I want to generate a Drep retirement (unregistration) certificate  So that I can submit it on a transaction and can get my DRep deposit back. | The command allows the user to provide the DRep credentials in the following ways:  **Using** the --drep-verification-key STRING option to specify the DRep verification key directly as a string.  **Using** the --drep-verification-key-file FILE option to specify the file containing the DRep verification key.  **Using** the --drep-id STRING option to specify the DRep ID directly as a string.  cardano-cli conway governance drep retirement-certificate | Typing cardano-cli conway governance drep retirement-certificate  with accepted input parameters will generate a DRep retirement certificate. |  |  |  |  |
| The command has the mandatory flag --deposit-amt to require the user to provide the drep deposited amount that is to be returned.   Must match the deposit originally paid when registering as DRep but is only checked when submitting the transaction. | The flag --deposit-amt is mandatory and takes the deposit amount in lovelace as an argument. The argument is a decimal, providing any other type of input will result in an exception (no implicit conversion)  Not providing the flag and its argument generates an exception. |  |  |  |  |
| The --out-file FILE option should be mandatory and used to specify the file where the generated DRep retirement certificate will be saved | The flag --out-file is mandatory and takes the file path and name as an argument.  Failure to provide the flag and its argument generates an exception. |  |  |  |  |
| The certificate should be on a text envelope format, similar to what we have for stake pools deregistration certificates | Given that the certificate is saved, then it is in a text envelope format consisting of a json object with type, description and cbor hex fields, where:  {  "type": "CertificateConway",  "description": "DRep Retirement Certificate",  "cborHex": "00000000"  }   |  |  |  |  |
| The output certificate complies with the conway cddl   unreg\_drep\_cert = (17, drep\_credential, coin) | **Detail Pending** |  |  |  |  |
| The feature implementation should be well-documented, providing clear usage instructions | **Detail Pending** |  |  |  |  |
| The command handles errors gracefully and provides helpful error messages when required options are missing or invalid inputs are provided. | **Detail Pending** |  |  |  |  |
| CLI12 | As a Drep  I want to generate the hash of my DRep metadata  So that I can supply it when registering as DRep | The command calculates the blake2b 256 hash of the file supplied by the user.  cardano-cli conway governance drep metadata-hash | **Detail Pending** |  |  |  |  |
| The command requires the --drep-metadata-file FILE option to specify the file containing the DRep metadata. | **Detail Pending** |  |  |  |  |
| The command supports the --out-file FILE option (optional) to enable users to save the calculated metadata hash to the specified file. If the flag is not used, the hash is printed to stdout | **Detail Pending** |  |  |  |  |
| The command handles errors gracefully and provides helpful error messages when required options are missing or invalid inputs are provided. | **Detail Pending** |  |  |  |  |
| CLI13 | As an ada holder  I want to create a governance action that updates the constitution  So that it can be submitted to the chain and be voted by the governance bodies | The command requires either --mainnet or --testnet-magic NATURAL option to specify the target network for which the governance action is generated  cardano-cli conway governance action create-constitution | **Detail Pending** |  |  |  |  |
| The command requires the user to provide the the deposit amount for submitting governance actions via the flag *--governance-action-deposit* | **Detail Pending** |  |  |  |  |
| The command requires the user to provide the stake credential that will receive the deposit return when the action is enacted/expired. It accepts:  --stake-verification-key-file  --stake-verification-key  --stake-key-hash | **Detail Pending** |  |  |  |  |
| The command allows the user to provide the transaction id and index of the previously enacted action of this type. These flags are optional (but if one is used the other one must be used too) to support the very first action of this type on the system which does not require information about previously enacted actions. The flags are:  --governance-action-tx-id  --governance-action-index | **Detail Pending** |  |  |  |  |
| The command requires the user to provide an anchor (url / hash) of the proposal. A document where the proposer exposes the reasoning behind the proposed change.  --anchor-url  --anchor-data-hash | **Detail Pending** |  |  |  |  |
| The command requires the user to provide an anchor (url / hash) of the new constitution.  --constitution-url  --constitution-hash | **Detail Pending** |  |  |  |  |
| The command has the mandatory flag –out-file to specify the file here the governance action (the proposal) will be saved. | **Detail Pending** |  |  |  |  |
| The generated governance action complies with the conway cddl, where:  proposal\_procedure =  [ deposit : coin  , reward\_account  , gov\_action  , anchor  ]    new\_constitution = (5, gov\_action\_id / null, constitution)    constitution =  [ anchor  , scripthash / null  ] | **Detail Pending** |  |  |  |  |
| CLI14 | As an ada holder  I want to create a governance action that updates the constitutional committee  So that it can be submitted to the chain and be voted by the governance bodies | The command requires either --mainnet or --testnet-magic NATURAL option to specify the target network for which the action is generated | **Detail Pending** |  |  |  |  |
| The command requires the user to provide the the deposit amount for submitting governance actions via the flag *--governance-action-deposit* | **Detail Pending** |  |  |  |  |
| The command requires the user to provide the stake credential that will receive the deposit return when the action is enacted/expired. It accepts:  --stake-verification-key-file  --stake-verification-key  --stake-key-hash | **Detail Pending** |  |  |  |  |
| The command requires the user to provide an anchor (url / hash) of the proposal. A document where the proposer exposes the reasoning behind the proposed change.  --anchor-url  --anchor-data-hash | **Detail Pending** |  |  |  |  |
| The command offers the option to remove many CC members, it accepts:  --remove-cc-cold-verification-key-hash  --remove-cc-cold-verification-key  --remove-cc-cold-verification-key-file | **Detail Pending** |  |  |  |  |
| The command offers the option to add many CC members.  --add-cc-cold-verification-key  --add-cc-cold-verification-key-file  --add-cc-cold-verification-key-hash | **Detail Pending** |  |  |  |  |
| When adding a new member, the command requires the user to also provide a **term** for each new member using the flag **–epoch** | **Detail Pending** |  |  |  |  |
| The command allows proposing a new quorum threshold:    **When** adding members  **When** removing members  **As** standalone action (no adds or removals) | **Detail Pending** |  |  |  |  |
| The command has the flag –out-file to specify the file where the governance action (the proposal) will be saved. | **Detail Pending** |  |  |  |  |
| The generated governance action complies with the conway cddl, where:    proposal\_procedure =  [ deposit : coin  , reward\_account  , gov\_action  , anchor  ]      update\_committee = (4, gov\_action\_id / null, set<committee\_cold\_credential>, { committee\_cold\_credential => epoch }, unit\_interval) | **Detail Pending** |  |  |  |  |
| CLI15 | As an ada holder  I want to create a governance action to withdraw funds from the treasury  So that it can be submitted to the chain and be voted by the governance bodies  cardano-cli conway governance action create-treasury-withdrawal | The command requires either --mainnet or --testnet-magic NATURAL option to specify the target network for which the action is generated | **Detail Pending** |  |  |  |  |
| The command requires the user to provide the the deposit amount for submitting governance actions via the flag *--governance-action-deposit* | **Detail Pending** |  |  |  |  |
| The command requires the user to provide the stake credential that will receive the deposit return when the action is enacted/expired. It accepts:  --stake-verification-key-file  --stake-verification-key  --stake-key-hash | **Detail Pending** |  |  |  |  |
| The command requires the user to provide an anchor (url / hash) of the proposal. A document where the proposer exposes the reasoning behind the proposed change.  --anchor-url  --anchor-data-hash | **Detail Pending** |  |  |  |  |
| The command requires the user to provide the stake credential that will receive the funds if the governance action is ratified. It accepts:  --stake-verification-key-file  --stake-verification-key  --stake-key-hash | **Detail Pending** |  |  |  |  |
| Requires the user to provide the **amount** in lovelace that will be transferred from the treasury to the stake credential if the action is ratified. | **Detail Pending** |  |  |  |  |
| The command has the flag –out-file to specify the file where the governance action (the proposal) will be saved. | **Detail Pending** |  |  |  |  |
| The generated governance action complies with the conway cddl, where:    proposal\_procedure =  [ deposit : coin  , reward\_account  , gov\_action  , anchor  ]    treasury\_withdrawals\_action = (2, { reward\_account => coin }) | **Detail Pending** |  |  |  |  |
| CLI16 | As an ada holder  I want to create an info governance action  So that it can be submitted to the chain and be voted by the governance bodies  cardano-cli conway governance action create-info | The command requires either --mainnet or --testnet-magic NATURAL option to specify the target network for which the action is generated. | **Detail Pending** |  |  |  |  |
| The command requires the user to provide the the deposit amount for submitting governance actions via the mandatory flag *--governance-action-deposit* | **Detail Pending** |  |  |  |  |
| The command requires the user to provide the stake credential that will receive the deposit return when the action is enacted/expired. It accepts:  --stake-verification-key-file  --stake-verification-key  --stake-key-hash | **Detail Pending** |  |  |  |  |
| The command requires the user to provide an anchor (url / hash) of the proposal. A document where the proposer exposes the reasoning behind the proposed change.  --anchor-url  --anchor-data-hash | **Detail Pending** |  |  |  |  |
| The command has the flag –out-file to specify the file where the governance action (the proposal) will be saved. | **Detail Pending** |  |  |  |  |
| The generated governance action complies with the conway cddl, where:    proposal\_procedure =  [ deposit : coin  , reward\_account  , gov\_action  , anchor  ]    info\_action = 6 | **Detail Pending** |  |  |  |  |
| CLI17 | As an ada holder  I want to create a governance action to update protocol parameters  So that it can be submitted to the chain and be voted by the governance bodies  cardano-cli conway governance action create-protocol-parameters-update | The command requires either --mainnet or --testnet-magic NATURAL option to specify the target network for which the DRep registration certificate is generated. | **Detail Pending** |  |  |  |  |
| The command requires the user to provide the the deposit amount for submitting governance actions via the mandatory flag *--governance-action-deposit* | **Detail Pending** |  |  |  |  |
| The command requires the user to provide the stake credential that will receive the deposit return when the action is enacted/expired. It accepts:  --stake-verification-key-file  --stake-verification-key  --stake-key-hash | **Detail Pending** |  |  |  |  |
| The command allows the user to provide the transaction id and index of the previously enacted action of this type. These flags are optional (but if one is used the other one must be used too) to support the very first action of this type on the system which does not require information about previously enacted actions. The flags are:  --governance-action-tx-id  --governance-action-index | **Detail Pending** |  |  |  |  |
| The command requires the user to provide an anchor (url / hash) of the proposal. A document where the proposer exposes the reasoning behind the proposed change.  --anchor-url  --anchor-data-hash | **Detail Pending** |  |  |  |  |
| The command includes dedicated flags to reference the protocol parameter that the user is attempting to modify. The parameters that can be included in this type of proposal are:  The **network group** consists of:  maximum block body size (maxBBSize)  maximum transaction size (maxTxSize)  maximum block header size (maxBHSize)  maximum size of a serialized asset value (maxValSize)  maximum script execution units in a single transaction (maxTxExUnits)  maximum script execution units in a single block (maxBlockExUnits)  maximum number of collateral inputs (maxCollateralInputs)  The **economic group** consists of:  minimum fee coefficient (minFeeA)  minimum fee constant (minFeeB)  delegation key Lovelace deposit (keyDeposit)  pool registration Lovelace deposit (poolDeposit)  monetary expansion (rho)  treasury expansion (tau)  minimum fixed rewards cut for pools (minPoolCost)  minimum Lovelace deposit per byte of serialized UTxO (coinsPerUTxOByte)  prices of Plutus execution units (prices)  The **technical group** consists of:  pool pledge influence (a0)  pool retirement maximum epoch (eMax)  desired number of pools (nOpt)  Plutus execution cost models (costModels)  proportion of collateral needed for scripts (collateralPercentage)  The **governance group** consists of all the new protocol parameters that are introduced in this CIP:  governance voting thresholds  dRepVotingThresholds  dvtCommitteeNoConfidence  dvtCommitteeNormal  dvtHardForkInitiation  dvtMotionNoConfidence  dvtPPEconomicGroup  dvtPPGovGroup  dvtPPNetworkGroup  dvtPPTechnicalGroup  dvtTreasuryWithdrawal  dvtUpdateToConstitution  poolVotingThresholds  pvtCommitteeNoConfidence  pvtCommitteeNormal  pvtHardForkInitiation  pvtMotionNoConfidence  governance action maximum lifetime in epochs (govActionLifetime)  governance action deposit (govActionDeposit)  DRep deposit amount (drepDeposit)  DRep activity period in epochs (drepActivity)  minimal constitutional committee size (ccMinSize)  maximum term length (in epochs) for the constitutional committee members (ccMaxTermLength) | **Detail Pending** |  |  |  |  |
| The command has the flag –out-file to specify the file where the governance action (the proposal) will be saved. | **Detail Pending** |  |  |  |  |
| The generated governance action complies with the conway cddl, where:    proposal\_procedure =  [ deposit : coin  , reward\_account  , gov\_action  , anchor  ]    parameter\_change\_action = (0, gov\_action\_id / null, protocol\_param\_update) | **Detail Pending** |  |  |  |  |
| CLI18 | As an ada holder  I want to create a no-confidence governance action  So that it can be submitted to the chain and be voted by the governance bodies  cardano-cli conway governance action create-no-confidence | The command requires either --mainnet or --testnet-magic NATURAL option to specify the target network for which the action is generated. | **Detail Pending** |  |  |  |  |
| The command requires the user to provide the the deposit amount for submitting governance actions via the mandatory flag *--governance-action-deposit* | **Detail Pending** |  |  |  |  |
| The command requires the user to provide the stake credential that will receive the deposit return when the action is enacted/expired. It accepts:  --stake-verification-key-file  --stake-verification-key  --stake-key-hash | **Detail Pending** |  |  |  |  |
| The command allows the user to provide the transaction id and index of the previously enacted action of this type. These flags are optional (but if one is used the other one must be used too) to support the very first action of this type on the system which does not require information about previously enacted actions. The flags are:  --governance-action-tx-id  --governance-action-index | **Detail Pending** |  |  |  |  |
| The command requires the user to provide an anchor (url / hash) of the proposal. A document where the proposer exposes the reasoning behind the proposed change.  --anchor-url  --anchor-data-hash | **Detail Pending** |  |  |  |  |
| The command has the flag –out-file to specify the file where the governance action (the proposal) will be saved. | **Detail Pending** |  |  |  |  |
| The generated governance action complies with the conway cddl, where:    proposal\_procedure =  [ deposit : coin  , reward\_account  , gov\_action  , anchor  ]    no\_confidence = (3, gov\_action\_id / null) | **Detail Pending** |  |  |  |  |
| CLI19 | As an ada holder  I want to create a governance action to initiate a hardfork  So that it can be submitted to the chain and be voted by the governance bodies | The command requires either --mainnet or --testnet-magic NATURAL option to specify the target network for which the action is generated. | **Detail Pending** |  |  |  |  |
| The command requires the user to provide the the deposit amount for submitting governance actions via the mandatory flag *--governance-action-deposit* | **Detail Pending** |  |  |  |  |
| The command requires the user to provide the stake credential that will receive the deposit return when the action is enacted/expired. It accepts:  --stake-verification-key-file  --stake-verification-key  --stake-key-hash | **Detail Pending** |  |  |  |  |
| The command allows the user to provide the transaction id and index of the previously enacted action of this type. These flags are optional (but if one is used the other one must be used too) to support the very first action of this type on the system which does not require information about previously enacted actions. The flags are:  --governance-action-tx-id  --governance-action-index | **Detail Pending** |  |  |  |  |
| The command requires the user to provide an anchor (url / hash) of the proposal. A document where the proposer exposes the reasoning behind the proposed change.  --anchor-url  --anchor-data-hash | **Detail Pending** |  |  |  |  |
| The command requires the user to input the **new protocol version number** | **Detail Pending** |  |  |  |  |
| The command has the flag –out-file to specify the file where the governance action (the proposal) will be saved. | **Detail Pending** |  |  |  |  |
| The generated governance action complies with the conway cddl, where:    proposal\_procedure =  [ deposit : coin  , reward\_account  , gov\_action  , anchor  ]    hard\_fork\_initiation\_action = (1, gov\_action\_id / null, [protocol\_version]) | **Detail Pending** |  |  |  |  |
| CLI20 | As an ada holder  I want to inspect the contents of a governance action file  So that I can verify it is correct before submitting it in a transaction  cardano-cli conway governance action view | The command takes an action file as input | **Detail Pending** |  |  |  |  |
| Gives an option to select the output format (json or yaml) | **Detail Pending** |  |  |  |  |
| The command has the flag –out-file to specify the file where the output is saved, if it is not specified it is printed to stdout | **Detail Pending** |  |  |  |  |
| The output shows the information of the proposal based on the proposal procedures on a human readable format:  proposal\_procedure =  [ deposit : coin  , reward\_account  , gov\_action  , anchor  ] | **Detail Pending** |  |  |  |  |
| CLI21 | As a Drep, SPO or CC member  I want to create a vote for a governance action  So that I can include it in a transaction and submit it to the chain  cardano-cli conway governance vote create | The command provides a way to vote  yes,  no  abstain | **Detail Pending** |  |  |  |  |
| Requires to specify the governance action ID and index that the vote is about. | **Detail Pending** |  |  |  |  |
| Requires the user to provide DRep, SPO or CC credentials | **Detail Pending** |  |  |  |  |
| The command allows the user to provide an optional anchor (url / hash) if the user wishes to share the reasoning behind the vote.  --anchor-url  --anchor-data-hash | **Detail Pending** |  |  |  |  |
| The command has the flag –out-file to specify the file where the vote will be saved. | **Detail Pending** |  |  |  |  |
| The generated vote complies with the conway cddl where   voting\_procedures = { + voter => { + gov\_action\_id => voting\_procedure } }    voting\_procedure =  [ vote  , anchor / null  ]    ; no - 0  ; yes - 1  ; abstain - 2  vote = 0 .. 2 | **Detail Pending** |  |  |  |  |
| CLI22 | As a DRep, SPO or CC member  I want to inspect the contents of a vote file  So that I can verify it is correct before submitting it in a transaction  cardano-cli conway governance vote view | The command takes a vote file as an input | **Detail Pending** |  |  |  |  |
| Gives an option to select the output format (json or yaml) | **Detail Pending** |  |  |  |  |
| The command has the flag –out-file to specify the file where the output is saved, if it is not specified it is printed to stdout | **Detail Pending** |  |  |  |  |
| The output shows the information of the proposal based on the voting procedures in a human readable format (english):    voting\_procedures = { + voter => { + gov\_action\_id => voting\_procedure } }    voting\_procedure =  [ vote  , anchor / null  ]    ; no - 0  ; yes - 1  ; abstain - 2  vote = 0 .. 2 | **Detail Pending** |  |  |  |  |
| CLI23 | As an ada holder  I want to build a transaction that includes a proposal (containing a governance action)  So that I can later sign and submit to the chain  transaction build | Transaction **build** has a new flag to supply a proposal file as input for the transaction body | **Detail Pending** |  |  |  |  |
| When constructing a transaction body that includes a proposal, the resulting tx body conforms to the conway cddl so that proposal procedures are recorded with the key 20  transaction\_body =  { 0 : set<transaction\_input> ; inputs  , 1 : [\* transaction\_output]  , 2 : coin ; fee  , ? 3 : uint ; time to live  , ? 4 : certificates  , ? 5 : withdrawals  , ? 7 : auxiliary\_data\_hash  , ? 8 : uint ; validity interval start  , ? 9 : mint  , ? 11 : script\_data\_hash  , ? 13 : nonempty\_set<transaction\_input> ; collateral inputs  , ? 14 : required\_signers  , ? 15 : network\_id  , ? 16 : transaction\_output ; collateral return  , ? 17 : coin ; total collateral  , ? 18 : nonempty\_set<transaction\_input> ; reference inputs  , ? 19 : voting\_procedures ; New; Voting procedures  **, ? 20 : proposal\_procedures ; New; Proposal procedures**  , ? 21 : coin ; New; current treasury value  , ? 22 : positive\_coin ; New; donation  } | **Detail Pending** |  |  |  |  |
| CLI24 | As a DRep, SPO or CC member  I want to build a transaction that includes my vote on a particular governance action  So that I can later sign and submit to the chain  transaction build | Transaction **build** has a new flag to supply a vote file as input for the transaction body | **Detail Pending** |  |  |  |  |
| When constructing a transaction body that includes a vote, the resulting tx body conforms to the conway cddl so that voting procedures are recorded with the key 19  transaction\_body =  { 0 : set<transaction\_input> ; inputs  , 1 : [\* transaction\_output]  , 2 : coin ; fee  , ? 3 : uint ; time to live  , ? 4 : certificates  , ? 5 : withdrawals  , ? 7 : auxiliary\_data\_hash  , ? 8 : uint ; validity interval start  , ? 9 : mint  , ? 11 : script\_data\_hash  , ? 13 : nonempty\_set<transaction\_input> ; collateral inputs  , ? 14 : required\_signers  , ? 15 : network\_id  , ? 16 : transaction\_output ; collateral return  , ? 17 : coin ; total collateral  , ? 18 : nonempty\_set<transaction\_input> ; reference inputs  **, ? 19 : voting\_procedures ; New; Voting procedures**  , ? 20 : proposal\_procedures ; New; Proposal procedures  , ? 21 : coin ; New; current treasury value  , ? 22 : positive\_coin ; New; donation  } | **Detail Pending** |  |  |  |  |
| CLI25 | As an ada holder  I want to build a transaction that includes a proposal (containing a governance action)  So that I can later sign and submit to the chain  transaction build-raw | Transaction build-raw has a new flag to supply a proposal file as input for the transaction body | **Detail Pending** |  |  |  |  |
| When constructing a transaction body that includes a proposal, the resulting tx body conforms to the conway cddl so that proposal procedures are recorded with the key 20  transaction\_body =  { 0 : set<transaction\_input> ; inputs  , 1 : [\* transaction\_output]  , 2 : coin ; fee  , ? 3 : uint ; time to live  , ? 4 : certificates  , ? 5 : withdrawals  , ? 7 : auxiliary\_data\_hash  , ? 8 : uint ; validity interval start  , ? 9 : mint  , ? 11 : script\_data\_hash  , ? 13 : nonempty\_set<transaction\_input> ; collateral inputs  , ? 14 : required\_signers  , ? 15 : network\_id  , ? 16 : transaction\_output ; collateral return  , ? 17 : coin ; total collateral  , ? 18 : nonempty\_set<transaction\_input> ; reference inputs  , ? 19 : voting\_procedures ; New; Voting procedures  **, ? 20 : proposal\_procedures ; New; Proposal procedures**  , ? 21 : coin ; New; current treasury value  , ? 22 : positive\_coin ; New; donation  } | **Detail Pending** |  |  |  |  |
| CLI26 | As a DRep, SPO or CC member  I want to build a transaction that includes my vote on a particular governance action  So that I can later sign and submit to the chain  transaction build-raw | Transaction build-raw has a new flag to supply a vote file as input for the transaction body | **Detail Pending** |  |  |  |  |
| When constructing a transaction body that includes a vote, the resulting tx body conforms to the conway cddl so that voting procedures are recorded with the key 19  transaction\_body =  { 0 : set<transaction\_input> ; inputs  , 1 : [\* transaction\_output]  , 2 : coin ; fee  , ? 3 : uint ; time to live  , ? 4 : certificates  , ? 5 : withdrawals  , ? 7 : auxiliary\_data\_hash  , ? 8 : uint ; validity interval start  , ? 9 : mint  , ? 11 : script\_data\_hash  , ? 13 : nonempty\_set<transaction\_input> ; collateral inputs  , ? 14 : required\_signers  , ? 15 : network\_id  , ? 16 : transaction\_output ; collateral return  , ? 17 : coin ; total collateral  , ? 18 : nonempty\_set<transaction\_input> ; reference inputs  **, ? 19 : voting\_procedures ; New; Voting procedures**  , ? 20 : proposal\_procedures ; New; Proposal procedures  , ? 21 : coin ; New; current treasury value  , ? 22 : positive\_coin ; New; donation  } | **Detail Pending** |  |  |  |  |
| CLI27 | As an ada holder  I want to create a conway cddl-compliant stake registration certificate  stake-address | Allows the user to provide credentials in any of the following forms:    Stake verification key  Stake verification key file  Stake address  Stake script file | **Detail Pending** |  |  |  |  |
| Requires the user to provide the required key deposit in lovelace | **Detail Pending** |  |  |  |  |
| The command has the flag –out-file to specify the file where the registration certificate will be saved. | **Detail Pending** |  |  |  |  |
| The command handles errors gracefully and provides helpful error messages when required options are missing or invalid inputs are provided. | **Detail Pending** |  |  |  |  |
| The certificate should be saved on a text envelope format:  {  "type": "CertificateConway",  "description": "Stake Address Registration Certificate",  "cborHex": ""  }   | **Detail Pending** |  |  |  |  |
| The resulting certificate conforms with the conway cddl, where   **reg\_cert = (7, stake\_credential, coin)** | **Detail Pending** |  |  |  |  |
| CLI28 | As an ada holder  I want to create a conway cddl-compliant stake deregistration certificate to get my deposit back  stake-address | Allows the user to provide credentials in any of the following forms:    Stake verification key  Stake verification key file  Stake address  Stake script file | **Detail Pending** |  |  |  |  |
| Requires the user to provide key deposit (in lovelace) that was paid by the target stake credential when it registered | **Detail Pending** |  |  |  |  |
| The command has the flag –out-file to specify the file where the registration certificate will be saved. | **Detail Pending** |  |  |  |  |
| The command handles errors gracefully and provides helpful error messages when required options are missing or invalid inputs are provided. | **Detail Pending** |  |  |  |  |
| The certificate should be saved on a text envelope format:  {  "type": "CertificateConway",  "description": "Stake Address Deregistration Certificate",  "cborHex": ""  }   | **Detail Pending** |  |  |  |  |
| The resulting certificate conforms with the conway cddl, where   **unreg\_cert = (8, stake\_credential, coin)** | **Detail Pending** |  |  |  |  |
| CLI29 | As an ada holder  I want to delegate my votes to a DRep (registered or default)  So that my stake is counted when the DRep vote.  Stake-address | Allows the user to provide credentials in any of the following forms:    Stake verification key  Stake verification key file  Stake address  Stake script file | **Detail Pending** |  |  |  |  |
| When delegating to a registered DRep, the user can provide the target Drep with:  DRep script hash  DRep verification key  DRep verification key file  DRep key hash (Drep ID) | **Detail Pending** |  |  |  |  |
| When delegating to a default DRep the user can use a flag to select either always-abstain or always-no-confidence, but not both. | **Detail Pending** |  |  |  |  |
| The command has the flag –out-file to specify the file where the vote delegation certificate will be saved. | **Detail Pending** |  |  |  |  |
| The command handles errors gracefully and provides helpful error messages when required options are missing or invalid inputs are provided. | **Detail Pending** |  |  |  |  |
| The certificate should be saved on a text envelope format:  {  "type": "CertificateConway",  "description": "Vote Delegation Certificate",  "cborHex": ""  }   | **Detail Pending** |  |  |  |  |
| The resulting certificate conforms with the conway cddl, where   **vote\_deleg\_cert = (9, stake\_credential, drep)** | **Detail Pending** |  |  |  |  |
| CLI30 | As an ada holder  I want to delegate my stake to a stake pool AND my votes to a DRep (registered or default) with a single certificate.  stake-address | Allows the user to provide credentials in any of the following forms:    Stake verification key  Stake verification key file  Stake address  Stake script file | **Detail Pending** |  |  |  |  |
| The user can provide the target stake pool with:  Stake pool cold verification key  Stake pool cold verification key file  Stake pool ID | **Detail Pending** |  |  |  |  |
| Allows the user to provide the target DRep credential with    DRep script hash  DRep verification key  DRep verification key file  DRep key hash (Drep ID) | **Detail Pending** |  |  |  |  |
| When delegating to a default DRep the user can use a flag to select either always-abstain or always-no-confidence, but not both. | **Detail Pending** |  |  |  |  |
| The command has the flag –out-file to specify the file where the stake and vote delegation certificate will be saved. | **Detail Pending** |  |  |  |  |
| The command handles errors gracefully and provides helpful error messages when required options are missing or invalid inputs are provided. | **Detail Pending** |  |  |  |  |
| The certificate should be saved on a text envelope format:  {  "type": "CertificateConway",  "description": "Stake and Vote Delegation Certificate",  "cborHex": ""  }     | **Detail Pending** |  |  |  |  |
| The resulting certificate conforms with the conway cddl, where   **stake\_vote\_deleg\_cert = (10, stake\_credential, pool\_keyhash, drep)** | **Detail Pending** |  |  |  |  |
| CLI31 | As any persona  I want to query the nodes for the current  Governance state  So that I can inform my decisions | The new command is implemented as cardano-cli conway query gov-state | **Detail Pending** |  |  |  |  |
| The command requires the user to specify the network id (mainnet or testnet magic) | **Detail Pending** |  |  |  |  |
| The command has the flag –out-file to specify the file where the output will be saved. | **Detail Pending** |  |  |  |  |
| The output is a JSON showing, at least, the following information:    **Previous** protocol parameters  **Current** protocol parameters  **Last** enacted governance actions IDs for committee, constitution, hardfork initiation and parameter updates  **Ratified** governance actions that will be enacted at next epoch transition  **Active** governance actions proposals including their type, stake address that collects the deposit, governance action id, expiration, spo, drep and cc votes | **Detail Pending** |  |  |  |  |
| The command handles errors gracefully and provides helpful error messages when required options are missing or invalid inputs are provided. | **Detail Pending** |  |  |  |  |
| CLI32 | As a CC member  I want to query the committee state  So that I can find my expiration term and whether my hot key authorization certificate has been recorded on chain | The command is implemented as cardano-cli conway query committee-state | **Detail Pending** |  |  |  |  |
| Requires the user to provide the network id | **Detail Pending** |  |  |  |  |
| Supports a query for an specific CC credential  CC cold verification key  CC cold verification key file  CC cold verification key hash  CC hot verification key  CC hot verification key file  CC hot verification key hash | **Detail Pending** |  |  |  |  |
| When no CC key is specified, the command outputs information for all committee members | **Detail Pending** |  |  |  |  |
| The command allows filtering by active, expired and unrecognized members (registered hot keys to an unknown cc cold key) | **Detail Pending** |  |  |  |  |
| The output is a JSON showing, the following information:    Cold key hash  Hot credential status (Authorized/NotAuthorized)  When Authorized, shows the hot key hash  Status (Active, Expired, Unrecognized)  Expiration epoch  Current epoch (when you ran the query)  Quorum | **Detail Pending** |  |  |  |  |
| The command has the flag –out-file to specify the file where the output will be saved. | **Detail Pending** |  |  |  |  |
| The command handles errors gracefully and provides helpful error messages when required options are missing or invalid inputs are provided. | **Detail Pending** |  |  |  |  |
| CLI33 | As an ada holder  I want to query the DRep state  So that … | The command is implemented as cardano-cli conway query drep-state | **Detail Pending** |  |  |  |  |
| Requires the user to provide the network id | **Detail Pending** |  |  |  |  |
| Supports a query for an specific DRep credential    DRep verification key  DRep verification key file  DRep verification key hash (DRep ID) | **Detail Pending** |  |  |  |  |
| If no Drep credential is specified it returns all DReps | **Detail Pending** |  |  |  |  |
| The command has the flag –out-file to specify the file where the output will be saved. | **Detail Pending** |  |  |  |  |
| The output is a JSON showing, the following information:  Drep Key hash  Anchor (Drep metadata)  Deposit  Expiry (from Drep activity) | **Detail Pending** |  |  |  |  |
| The command handles errors gracefully and provides helpful error messages when required options are missing or invalid inputs are provided. | **Detail Pending** |  |  |  |  |
| CLI34 | As an ada holder and DRep  I want to query the DRep stake distribution  So that I can find the weight (of the votes) of each DRep | The command is implemented as cardano-cli conway query drep-stake-distribution | **Detail Pending** |  |  |  |  |
| Supports a query for an specific DRep credential    DRep verification key  DRep verification key file  DRep verification key hash (DRep ID) | **Detail Pending** |  |  |  |  |
| If no Drep credential is specified it returns all DReps | **Detail Pending** |  |  |  |  |
| The command has the flag –out-file to specify the file where the output will be saved. | **Detail Pending** |  |  |  |  |
| The output is a JSON showing, the following information:  Drep Key hash  Stake delegated to this DRep | **Detail Pending** |  |  |  |  |
| The command handles errors gracefully and provides helpful error messages when required options are missing or invalid inputs are provided. | **Detail Pending** |  |  |  |  |
| CLI35 | As an ada holder,  I want to query my stake address information so that I can learn to which pool and drep Im delegating to and the value in lovelace of my deposits for delegating and for submitting governance actions. | Expand the command query stake-address-info to return the drep id of the DRep that the stake credential is delegated to and the value of the existing deposits. |  |  |  |  |  |
|  | Script as a DRep |  |  |  |  |  |  |
| Sidechain User Stories | | | | | |  |  |
| CH.SID1 | As a Sidechain, I wish to use BLS primitives to record settlement operations on Cardano | [Jon Rossie](mailto:jon.rossie@iohk.io)posted your ‘thread comments’ to initiate | **Detail Pending** |  | **Detail Pending** | Sidechain |  |
| CH.SID2 | Sidechain will use BLS for tokenomics, in particular to support multi-sig certificates about Sidechain activity that can be verified by Plutus contracts | Detail Pending | **Detail Pending** |  | **Detail Pending** | Sidechain |  |
| Plutus Component Level User Stories | | | | | |  |  |
| CIP-85 | As a DApp developer I would like to use sums-of-products instead of Scott-encoding in my Plutus scripts to get better performance | Sums-of-products is available to use in Plutus V3 scripts and is the default way of encoding data types in Plutus Tx. | **Detail Pending** |  |  | Plutus |  |
| CIP-101 | User stories for Keccak256 primitive | Detail Pending | **Detail Pending** |  |  | Plutus |  |
|  | As a DApp developer I want to use the Blake2b-224 hashing function to compute PubKeyHash onchain | The Blake2b-224 is available to use in Plutus V3 after the HF | **Detail Pending** |  |  | Plutus |  |

# Revision & Approval

| **Name** | **Position** | **Role** | **Comments** | **Status** |
| --- | --- | --- | --- | --- |
| [Mike Ward](mailto:mike.ward@iohk.io) | Chief Product Officer | Approver |  | Under Review |
| [Michael Madoff](mailto:michael.madoff@iohk.io) | Product Manager - Voltaire | Approver |  | Under Review |
| [Samuel Leathers](mailto:samuel.leathers@iohk.io) | Product Manager - Core Tech | Approver |  | Under Review |
| [Dorin Solomon](mailto:dorin.solomon@iohk.io) | Quality Chapter Director | Approver |  |  |
| [Carlos Lopez de Lara](mailto:carlos.lopezdelara@iohk.io) | Product Owner - Core Tech | Reviewer |  | Under Review |
| [Kevin Hammond](mailto:kevin.hammond@iohk.io) | Head of Technology - Core Tech | Approver |  | Under Review |
| [Lorenzo Bruno](mailto:lorenzo.bruno@iohk.io) | Product Design Lead - Voltaire Governance Tools | Reviewer |  |  |
| [Thomas Upfield](mailto:thomas.upfield@iohk.io) | Product Owner - Voltaire | Reviewer |  | Under Review |
| [Gerard Moroney](mailto:gerard.moroney@iohk.io) | COO | Reviewer |  | Under Review |
| [Omer Husain](mailto:omer.husain@iohk.io) | Product Manager - Plutus | Reviewer | Confirm Smart Contracts User Stories | To Be Added |
| [Billy Mullins](mailto:billy.mullins@iohk.io) | VP Cardano | Reviewer | Confirm All User Stories | Under Review |
| [Dominik Zajkowski](mailto:dominik.zajkowski@iohk.io) | Sidechains Architect | Reviewer | Confirm Sidechains User Stories | Detail Pending |
| [Arnaud Bailly](mailto:arnaud.bailly@iohk.io) | Hydra Architect | Reviewer | Confirm Hydra User Stories |  |

# 