## Questions/analysis of changes

#### Links

osmosis/cosmos-sdk bank hooks: [link](https://github.com/osmosis-labs/cosmos-sdk/tree/sunny/bank-hooks)

**Branches:**

* <https://github.com/osmosis-labs/osmosis/tree/sunny/tokenfactory-addbackforce>
* <https://github.com/osmosis-labs/osmosis/tree/sunny/tokenfactory-beforesend>
* <https://github.com/osmosis-labs/osmosis/tree/sunny/tokenfactory-setmetadataq>
* <https://github.com/osmosis-labs/osmosis/tree/sunny/tokenfactory-blacklist> - THIS is not merged with fullpowered branch, seems that it would give me better context

**A:** This branch is second option in case that circle for USDC changes their mind and decides they wish to go with the blacklists defined in token factory module, not in the cosmwasm smart contract.

* <https://github.com/osmosis-labs/osmosis/tree/sunny/tokenfactory-supplyoffset> - probably not important?

**A:** no

Additional link, with an example of SmartContract:

<https://github.com/osmosis-labs/cw-usdc>

#### What would be the business use case?

A: for burning from and Forcing tx: If you wish to cash out your USDC to USD, Circle account will burn your desired USDC cash amount - if we do not have this feature, the user would need to send the amount to some other account, and the user would need to initiate the transaction of burning…  
This is more of the UX issues.

#### **What are osmosis fork Cosmos SDK changes compared to vanilla Cosmos SDK?** I have found in osmosis cosmos-sdk fork that for example:

* I have only found that governance and mint modules are explained in documentation online, but I don’t know if this is up to date.
* For the Governance module we have a change in EndBlocker methods when compared to vanilla governance module, but the change is implemented in forked version of SDK, and when it comes to Mint module it is entirely replaced with custom Mint module.

Example of one change, comparing to vanilla SDK module (if there is anything similar it could be important for the analysis of the new token factory impact):

* vanilla cosmos-sdk: when iterating through inactive proposals, in cosmos-sdk framework basic module we will be sending coins to the account that made a deposit, and delete deposits info
* Osmosis: there is an IsExpedited type of proposal, for those types we do not refund deposits if proposal fails,

iterating through inactive proposals, in osmosis we will be burning deposits and delete deposits info

**Q:** is there more changes that have been done in other modules?

* Is there an overview of these changes or a link to the documentation?
* Can anyone from the dev team walk us through the basic adaptations of framework you have had in order to get the behaviour?
* We need to analyze the wiring in osmosis’ app.go - order of basic modules and differences

**A:** Shelley will organize weekly or biweekly meetings with Dan.

**Note:** prior to starting the analysis we need to detect all the changes with vanilla modules, analysis would basically be drawing diagrams and analyzing the possible impact. This would be the first phase.

(Once we detect possible issues it would be necessary to recreate scenarios for testing if there would be issues with halting the chain.

For this step,it would be great if someone from your team could be a contact for setting up the environment for testing.)

## TokenFactory module

### Background

* To issue new tokens on Osmosis
* Cosmwasm exists on osmosis, where we have: CW20 standard (similar to ERC20 on Ethereum)
* Separation of Eth and ERC20 is a mistake - pain for integrators, wallets. Diverged code paths for different standards introduced large bugs!
* Solana does it weel.They have one contract that manages all tokens -sol, slp
* Osmosis wants to push tokenfactory as a standard instead of cw20.
* This is simple functionality inside of token factory.
* **Currently**: Admin can mint coins and then send them as normal bank SDK.coin and it can be transferred….
* **New features:** Limited functionality comparing to ERC20 and CW20 where contract can do more than just basic transfers. ERC20 - balance is what ever contract says it is. You can ForceTransfer, Burn….
* Blacklists: this address can send tokens or can’t be sent to an address - > blacklists can be implemented in cosmwasm smart contracts or in token factory module - Circle (USDC) will go with cosmwasm smart contract. Currently Circle is ok with the cosmwasm implementation of the hook. If they insist it will be moved to the token factory module.
* **Auditing of the cosmwasm smart contract** is not the focus now. Osmosis does not have any smart contracts deployed currently. It could be useful to get info about general stuff when using sudo exec of BeforeSendHook **Questions: DoS factors, gas limits, risks**, impact of smart contracts executing malicious things.

Uniswap and compound tokens . Transfers are tracked for 2 weeks and the gov proposals voting are done based on the snapshot of the voting power at the time the proposal is created. When you vote it rewinds your voting power to the time that proposal is made.

Re-basing feature: Osmosis wants to add additional features - except rebasing. Rebasing is when you re-denominate tokens automatically - you apply a multiplication to everyones balance.

Balances in ERC20 is defined by a result of a query - the rebalance is multiplying everyones balances.

Uniswap will as well reject this feature.