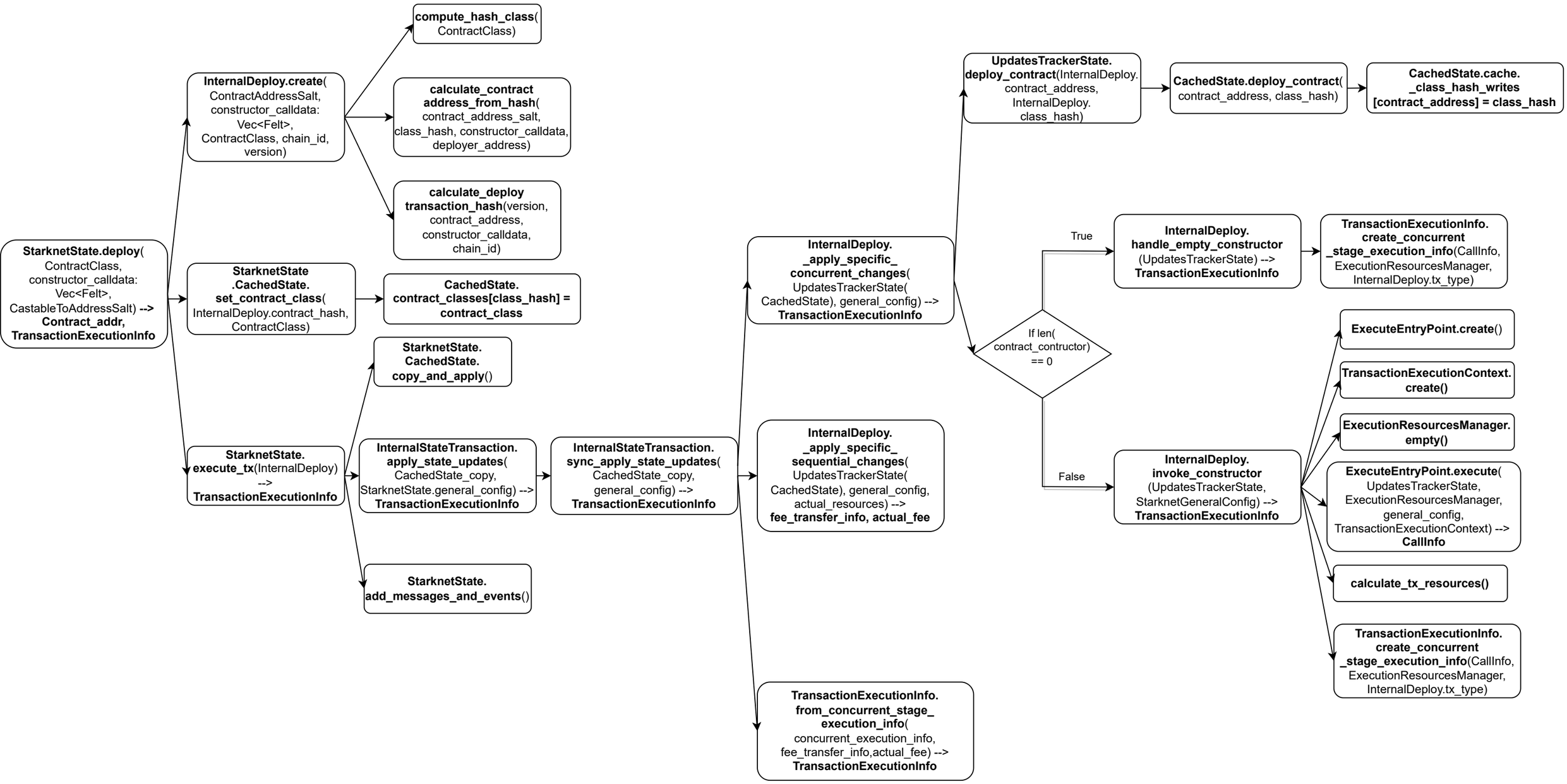
StarknetState.deploy( ContractClass, constructor_calldata: Vec<Felt>, CastableToAddressSalt) --> Contract_addr, TransactionExecutionInfo
InternalDeploy.create( ContractAddressSalt, constructor_calldata: Vec<Felt>, ContractClass, chain_id, version)
compute_hash_class( ContractClass)
calculate_contract address_from_hash( contract_address_salt, class_hash, constructor_calldata, deployer_address)
calculate_deploy transaction_hash(version, contract_address, constructor_calldata, chain_id)
StarknetState .CachedState. set_contract_class( InternalDeploy.contract_hash, ContractClass)
CachedState. contract_classes[class_hash] = contract_class
StarknetState. execute_tx(InternalDeploy) --> TransactionExecutionInfo
StarknetState. CachedState. copy_and_apply()
InternalStateTransaction. apply_state_updates( CachedState_copy, StarknetState.general_config) --> TransactionExecutionInfo
InternalStateTransaction. sync_apply_state_updates( CachedState_copy, general_config) --> TransactionExecutionInfo
StarknetState. add_messages_and_events()
InternalDeploy. _apply_specific_ concurrent_changes( UpdatesTrackerState( CachedState), general_config) --> TransactionExecutionInfo
InternalDeploy. _apply_specific_ sequential_changes( UpdatesTrackerState( CachedState), general_config, actual_resources) --> fee_transfer_info, actual_fee
TransactionExecutionInfo. from_concurrent_stage_ execution_info( concurrent_execution_info, fee_transfer_info,actual_fee) --> TransactionExecutionInfo
UpdatesTrackerState. deploy_contract(InternalDeploy. contract_address, InternalDeploy. class_hash)
CachedState.deploy_contract( contract_address, class_hash)
CachedState.cache. _class_hash_writes [contract_address] = class_hash
If len( contract_contructor) == 0
True
False
InternalDeploy. handle_empty_constructor (UpdatesTrackerState) --> TransactionExecutionInfo
TransactionExecutionInfo. create_concurrent _stage_execution_info(CallInfo, ExecutionResourcesManager, InternalDeploy.tx_type)
InternalDeploy. invoke_constructor (UpdatesTrackerState, StarknetGeneralConfig) --> TransactionExecutionInfo
ExecuteEntryPoint.create()
TransactionExecutionContext. create()
ExecutionResourcesManager. empty()
ExecuteEntryPoint.execute( UpdatesTrackerState, ExecutionResourcesManager, general_config, TransactionExecutionContext) --> CallInfo
calculate_tx_resources()
TransactionExecutionInfo. create_concurrent _stage_execution_info(CallInfo, ExecutionResourcesManager, InternalDeploy.tx_type)