# Ballot.sol

Used to vote for multiple proposals

Has a chairperson to add proposals

Winning proposal variable: **winningProposal**

# DemaxBallotFactory.sol

Used to vote for 2 proposals

Has a governor to end voting

Voter can delegate their vote to other voters, represented by “weight” and “delegate” in Voter struct

Winning proposal variable: **YES/NO**

# Gatekeeper.sol

Used to vote for multiple slates

Has a Gatekeeper to approve proposals

Uses tokens to vote

Each ballot has epoch number

Winning slate variable: **contest.winner**

# ibaVoter.sol

Used to vote for multiple proposals

Each ballot has a chainperson, used to identify the ballot

Winning proposal variable: **winner**

# localElection.sol

Originally planned for 2019 Hong Kong local elections

From source code:

Expected workflow, voter get his/her voter ID from contract, this makes sure no one has access to his/her personal data

Second, the voter registers their voter ID and email (not phone as no money to do SMS) via a vote box and make a vote.

Email is checked by the voter box to prevent double voting and robot voting, but there should be a better way to do it.

Now registration will make old registration and votes from the same voter ID invalid

The vote will then encrypted using a public key and submitted to this contract for record

After the election, the private key will be made public and people can decrypt the votes and knows who wins

Currently we let the vote box decides if a new vote should replace the old vote, but there should be a better way to do it

Also if people can read the private variable voteListByVoter from blockchain, they will know who a person votes after election.

The variable is needed for replacing old votes.

This variable should be removed when there is a proper way to authenticate a person and replacing vote is not needed.

Winning proposal variable: **councilVoterNumber[council]**

# PLCRVotingCheckpoint.sol

Used to vote for multiple proposals

Has ADMIN to grant permissions to create ballot

Winning proposal variable: winningProposal

# Redenom.sol

Vote for multiple projects

Has onlyAdmin to enable and disable voting

Winning project variable: **\_winningProject**

# TACVoting.sol

Used to vote for multiple matches

Only 1 election can be opened at once

Winning match variable: **Elections[electionId].winningMatch** / **winner**

# TomiQuery2.sol

Used to vote for 2 proposals

Has a governor to end voting

Winning proposal variable: **YES/NO**

# WeightedVoteCheckpoint.sol

Used to vote for multiple proposals

Winning proposal variables: **winningProposal**