MultiSigWallet With TimeLock:
The Smart Contract program serves to allow multiple parties to agree on transactions before execution. Once all of the parties confirm one transaction, the transaction will be executed by anyone in a required delay. In most cases, a relay server is responsible to poke to execute after the delay. In order to eliminate the need of relay server, we reimplemented the contracts in SigSolid and the transaction is executed in the form of signal transaction. In function `confirmTransaction()`, a signal event with transaction ID is emitted once all of the owners confirm. In the contract `UserExecuteTransaction` of a user, the handler function `multiSigWalletExecuation()` binds with the signal. After a required delay, the signal transaction of the user is executed and function `executeTransaction()` is invoked.

Historical Price Feed:
The contract is used to store updated historical price data from an off-chain oracle. The function `poke()` should be manually invoked every 24 hours in order to update prices periodically. In SigSolid, a signal event is declared without including any parameters. `poke()` is the corresponding handler function which emits the signal event inside. In this way, `poke()` signal transaction is always executed periodically to update the price data.

Idex:
The platform provides ERC-20 Token trading on Ethereum. Function `executeTrade()` in `Exchange` contract is responsible for settling trade orders. Once the chain propagation period has elapsed, `executeTrade()` invalidate all trade orders whose timestamp is lower than the one provided by calling `invalidateOrderNonce()`. In SigSolid, similar with Historical Price Feed, by setting empty signal emission in a handler function, order invalidations are executed every chain propagation period.

Aave:
Aave is decentralized lending pools protocol on Ethereum. Function `liquidationCall()` will be invoked by relay servers to check and liquidate undercollateralized position. There is a price update from oracle during the manual checking. In order to mitigate the unexpected delay caused by relay server poking, SigSolid introduces a periodical liquidation check mechanism similar with Historical Price Feed. In contract `LendingPoolLiquidationManager` of Aave, by setting empty signal emission in a handler function, the price is updated from oracle periodically. In the contract of users, the handler function binding with the empty signal will invoke `liquidationCall()` to check regularly.

Metronome:
Metronome is a DeFi market for a kind of ERC20 token on Ethereum. In contract `Metronome`, function `poke(address a)` is used to create a reward from the funds of address `a` who has not idled in the last 10 minutes. The user who invokes `invest()` in the last 10 minutes is regarded as Not-Idle status. In SigSolid, `poke(address a)` is set as a handler function. In `invest()`, a signal event with the sender address is emitted with the required delay. Then function `poke()` will be executed without the function call from reply servers. In addition, function `poke()` checks whether the user has pinged `invest()` during

the required period.

DAIHardFactory:
Contract DAIHardFactory provides a DAI trading platform. One trade process has five sequential phases. A fixed period called `"autoabortInterval"` is defined. During each phase, once `"autoabortInterval"` has passed, function `abort()` is executed by anyone to abort from the phase. Otherwise, the next phase will be executed. In SigSolid, at the beginning of each phase execution, a signal event with phase type is emitted. After a delay of `"autoabortInterval"`, the corresponding handler function first checks whether the current phase has changed or not. Next it aborts from the phase if the phase is not changed.

Contracts similar with MakerDAO:
Some contracts such as KyberDxMarketMaker provides price oracle service similar with MakerDAO. Users should use relay servers to call `poke()` to retrieve the latest price periodically. SigSolid solves the unexpected delay caused by relay servers.