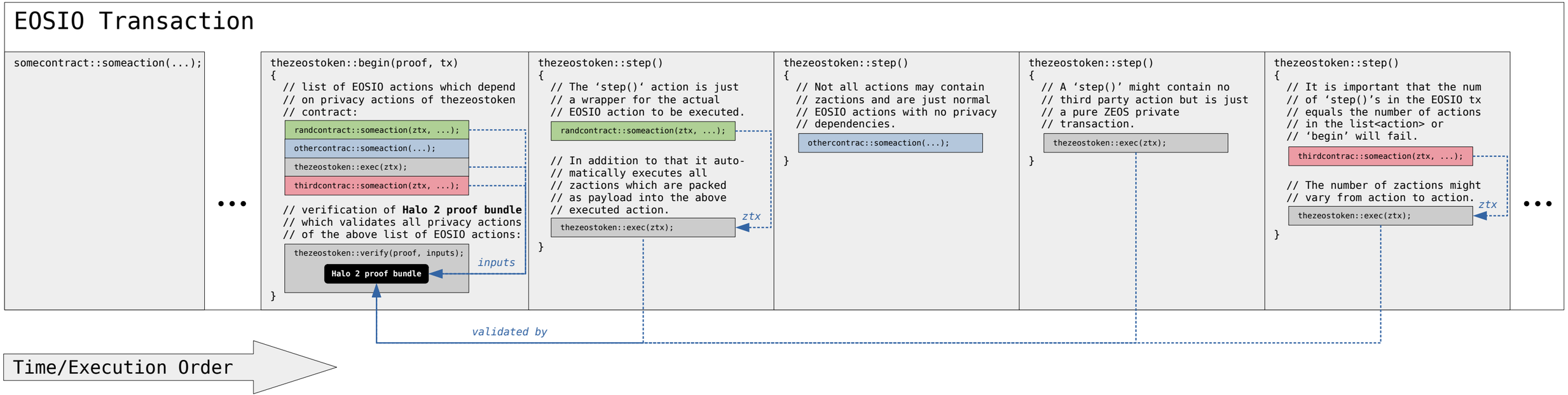
EOSIO Transaction
somecontract::someaction(...);
thezeostoken::begin(proof, tx)
{
// list of EOSIO actions which depend
// on privacy actions of thezeostoken
// contract:
randcontract::someaction(ztx, ...);
othercontrac::someaction(...);
thezeostoken::exec(ztx);
thirdcontrac::someaction(ztx, ...);
// verification of Halo 2 proof bundle
// which validates all privacy actions
// of the above list of EOSIO actions:
thezeostoken::verify(proof, inputs);
Halo 2 proof bundle
inputs
}
thezeostoken::step()
{
// The 'step()' action is just
// a wrapper for the actual
// EOSIO action to be executed.
randcontract::someaction(ztx, ...);
// In addition to that it auto-
// matically executes all
// zactions which are packed
// as payload into the above
// executed action.
thezeostoken::exec(ztx);
ztx
}
thezeostoken::step()
{
// Not all actions may contain
// zactions and are just normal
// EOSIO actions with no privacy
// dependencies.
othercontrac::someaction(...);
}
thezeostoken::step()
{
// A 'step()' might contain no
// third party action but is just
// a pure ZEOS private
// transaction.
thezeostoken::exec(ztx);
}
thezeostoken::step()
{
// It is important that the num
// of 'step()'s in the EOSIO tx
// equals the number of actions
// in the list<action> or
// 'begin' will fail.
thirdcontrac::someaction(ztx, ...);
// The number of zactions might
// vary from action to action.
thezeostoken::exec(ztx);
ztx
}
validated by
Time/Execution Order