

## Problem

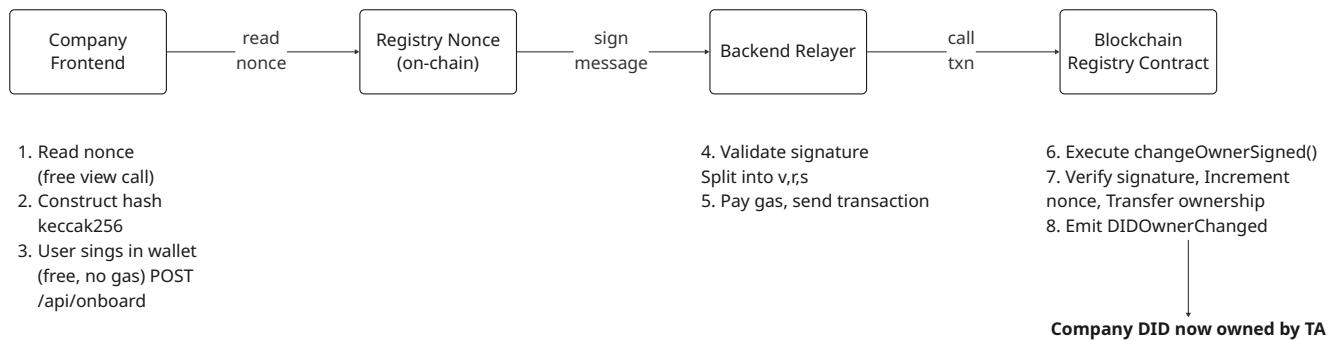
Our ecosystem requires companies to have funded crypto wallets before they can delegate control to the Trust Anchor. (Company admins also need to have funded wallets to make CRSet updates, but as a pilot application, we will be handling the company control delegation step here first.) This creates extra friction, and requires more interaction with the Blockchain, making it less accessible.

## Proposed Solution

Implement a **Gas Station relay service** that executes transactions on behalf of users, with the Trust Anchor covering gas costs. This would eliminate the need for companies (and companyAdmins later on) to hold crypto during onboarding

## Pilot Scope

Starting with the company onboarding flow as our first use case. When a company delegates DID control to the Trust Anchor, they currently must pay for this transaction. We leverage ERC-1056 / did:ethr contracts `changeOwnerSigned()` function, so companies can sign a message off-chain (free) while our backend relayer executes the transaction (TA pays gas)



## Alternative: ERC-4337 Account Abstraction

It would require Smart Contract Wallets, and higher complexity. It is also not compatible with DIDRegistry, as it requires an adapter. It looks promising, but is overkill for our current project.

## Implementation Requirements

- Backend service in demo-app-backend/ (Express + routes/controllers/services)
- Onboarding.tsx: Replace `useWriteContract` with signature flow (read nonce -> construct hash -> sign -> POST)
- contracts.ts: Add `changeOwnerSigned()`, `nonce()`, `DIDOwnerChanged` to `REGISTRY_ABI`
- Frontend .env - Add `VITE_BACKEND_URL`
- EthereumDIDRegistry (already has `changeOwnerSigned` at line 69)
- No new contract deployments or audits

## Testing:

- Create `GaslessOnboarding.test.ts`
- Test: successful delegation, replay protection, invalid signatures, "already-delegated" check