Group 6 — ThingID (SSI + IoT: device SBT + access passes)

One-liner: Issue **non-transferable device certificates** (SBT) to owners; mint **AccessPass** NFTs to authorize reading private IPFS streams until expiry.

A. Problem & Value

- IoT devices need verifiable identities and controllable data access. We bind devices to owners and grant **time-boxed, auditable** access to applications.

B. Functional Requirements

1. Issuer mints **DeviceCert SBT** to owner with device DID + public key.
2. Owner grants **AccessPass NFT** to an app/account with `expiresAt`.
3. API gateway checks AccessPass validity before proxying to IPFS private content.
4. UI shows who has access; owner can revoke passes.

Non-functional - Simple flows; minimal on-chain writes; no device secrets on-chain.

C. Architecture

```
[Issuer Admin]—mint SBT→[Contracts]
 [Owner DApp]—grant pass→[Contracts]
 [Viewer]—request→[Gateway]—check pass→[IPFS private]
```

- **Frontend**: Devices page, Access control, Viewer.
- **Backend**: Gateway enforcing AccessPass; IPFS private gateway proxy; pinner for metadata.
- **Contracts**: `DeviceCert` (SBT), `AccessPass` (ERC-721 with expiry map).

D. Data Models

D1) Device metadata (IPFS)

```
{ "did":"did:didlab:device-abc", "pubkey":"0x04...", "make":"SensorCo",
"model":"S1", "owner":"0x..." }
```

D2) Access log (off-chain, optional on-chain hash)

```
{ "deviceDid":"did:didlab:device-abc", "viewer":"0xViewer", "ts":1739577600,
"resource":"/stream/day/1" }
```

E. Smart Contracts (production-ready MVP)

```
// SPDX-License-Identifier: MIT
 pragma solidity ^0.8.21;
 import "@openzeppelin/contracts/token/ERC721/ERC721.sol";
 import
"@openzeppelin/contracts/token/ERC721/extensions/ERC721URIStorage.sol";
 import "@openzeppelin/contracts/access/AccessControl.sol";

contract DeviceCert is ERC721URIStorage, AccessControl {
   constructor() ERC721("DeviceCert","DCERT"){ _grantRole(DEFAULT_ADMIN_ROLE,
msg.sender);}
  function _beforeTokenTransfer(address from,address to,uint256 id,uint256)
internal override {
     require(from==address(0) || to==address(0), "SBT non-transferable");
     super._beforeTokenTransfer(from,to,id,1);
   }
   function mint(address to, uint256 id, string calldata uri) external
onlyRole(DEFAULT_ADMIN_ROLE){ _safeMint(to,id); _setTokenURI(id,uri);}
}

contract AccessPass is ERC721URIStorage, AccessControl {
   mapping(uint256=>uint64) public expiresAt; uint256 public nextId;
   constructor() ERC721("AccessPass","APASS"){ _grantRole(DEFAULT_ADMIN_ROLE,
msg.sender);}
  function mint(address to, string calldata uri, uint64 until) external
onlyRole(DEFAULT_ADMIN_ROLE) returns(uint256){ uint256 id=++nextId;
expiresAt[id]=until; _safeMint(to,id); _setTokenURI(id,uri); return id; }
   function valid(uint256 id) public view returns(bool){ return
block.timestamp < expiresAt[id]; }
 }
```

F. API

- `POST /issue-device` → pin device metadata, mint SBT, return tokenId
- `POST /grant-access` → mint AccessPass for viewer with expiry, return tokenId
- `GET /stream/:deviceId` → check caller's AccessPass ownership & validity, then proxy to IPFS private resource

G. Frontend UX

- **/devices**: list devices (owner view), details page with cert
- **/access**: grant/revoke passes, show who can view

- **/viewer**: open stream if pass valid; otherwise prompt to request

H. Day-by-Day Plan

5) Contracts + issuer admin

5) AccessPass + gateway checks

5) Frontend flows

5) IPFS private proxy + logs

5) Tests + docs + deploy

I. Testing Strategy

- SBT non-transferability
- Expiry enforcement (boundary times)
- Gateway refusal when no/expired pass
- Metadata correctness

J. Security & Privacy

- No device secrets on-chain; only public keys/DIDs
- Gateway is the only reader of IPFS private; logs accesses
- Principle of least privilege; role-gated mints

K. Deployment Steps

- Deploy both contracts; grant admin role to issuer key
- Configure gateway and IPFS credentials
- Hook front-end and tunnel

L. Seed & Demo

- Issue one device cert; grant 1-hour pass to viewer; demonstrate stream allowed/blocked across expiry

M. Docs

- Whitepaper: SSI for devices, pass model, operational concerns
- Deck: identity → grant → verify → expire demo

N. Stretch Goals

- Link with Group 4 SensorSeal data; automated pass creation on alerts; organization-level issuers