Integrating Blockchain into Solar-Powered Devices

1. Device & Sensor Layer
- Hardware components:
  - Solar inverter/meter with pulse or digital output (kWh counters)
  - Microcontroller/SoC (e.g. ESP32, Raspberry Pi Zero) to read meter pulses or Modbus data
  - Secure Element (e.g. ATECC608A) for on-device private key storage
- Data acquisition:
  - Sample energy production (e.g. every minute)
  - Aggregate into fixed intervals (e.g. 1 kWh buckets)
- Cryptographic signing:
  - Each interval's reading is signed by the device's private key
  - Signature + timestamp + device ID → signed data packet

2. Connectivity & Gateway
- Local networking:
  - Wi-Fi / Ethernet / LoRaWAN / cellular link to a local gateway or cloud endpoint
- Aggregation node:
  - Collect signed packets from multiple devices
  - Verify signatures against each device's public key
  - Batch into periodic on-chain transactions to save gas
- MQTT / HTTPS API:
  - Devices publish to an MQTT topic; gateway subscribes
  - Or devices POST to a REST endpoint

3. Blockchain Network Design
- Choice of chain:
  - Telkes PoW chain for full decentralization and direct TKL rewards
  - Or a Layer-2 / sidechain (e.g. Polygon, Energy Web Chain) for throughput
- Lightweight client or RPC:
  - Devices/gateways use an RPC node to submit transactions
  - Or embed a light client to verify block headers
- Smart Contracts:
  - EnergyRegistry contract stores:
    struct Report { address device; uint256 kWh; uint256 timestamp; }
    mapping(bytes32 => bool) processed; // replay protection
    function reportEnergy(bytes32 reportId, uint256 kWh, bytes signature) external { … }
  - Reward logic: on reportEnergy, mint or transfer kWh * 1 TKL to device owner, burning any fee portion

4. Token & Incentive Integration
- Device wallet:
  - Each device has its own on-chain address
  - Owner can sweep rewards from device into their personal wallet
- Gas abstraction:
  - Use meta-transactions so the gateway pays gas
  - Or embed a small amount of TKL on each device to fund transactions
- Batching & scaling:
  - Gateway groups multiple reports into one transaction
  - Saves fees and reduces on-chain congestion

5. Security & Trust
- Secure key storage ensures devices can't be spoofed
- Replay protection (nonce or processed[reportId]) prevents double-claiming
- Device onboarding via DAO-governed registry of approved device public keys
- Over-the-air updates to patch vulnerabilities

6. User Interface & Monitoring
- Dashboard:
  • Real-time charts of kWh reported vs. TKL rewarded
  • Device health & connectivity status
- Mobile App / Web Portal:
  • Let owners view, claim, or sell their rewards
  • Enable P2P energy trading: devices/owners place sell orders on-chain
Example Workflow:
1. Device A generates 5 kWh in one hour.
2. Controller signs {deviceID, 5, timestamp}.
3. Sends packet to Gateway via MQTT.
4. Gateway verifies and calls EnergyRegistry.reportEnergy(...) on Telkes.
5. Smart contract transfers 5 TKL to Device A's address (95% net, 5% burned).