## **A. Meta**

**Table**

| **Key** | **Value** |
| --- | --- |
| Version | 1.0.0-draft |
| Date | 2025-09-06 |
| ****Copyright**** | © 2025 Vilokanam-View. ****All Rights Reserved.**** |
| Repository | [https://github.com/Vilokanam-View/Vilokanam-View](https://github.com/vilokanam/vilokanam) |
| Docs URI | [https://docs.Vilokanam-View.io](https://docs.vilokanam.io/) |
| Issue Tracker | [https://github.com/Vilokanam-View/Vilokanam-View/issues](https://github.com/vilokanam/vilokanam/issues) |
| Hackathon | Polkadot “Build Resilient Apps” 6 Oct – 17 Nov 2025 |

## **B. 30-Second Elevator Pitch**

Vilokanam-View is a Polkadot parachain plus lightweight Web2 SDK that turns every second of live video or audio into a ****micro-payment**** (default $0.0007).  
Viewers lock funds for ****at most 60 s****; creators withdraw ****every minute**** and automatically earn ****staking yield on the float****.  
No charge-backs, no 50 % platform cut, no three-week payout delays—****Web2 UX with Web3 rails****.

## **C. Problem & Market Gap**

1. ****Creators****
   * Twitch: 50 % revenue share, 45-day hold.
   * OnlyFans: arbitrary freezes.
   * Zoom / Google Meet: ****zero**** direct monetisation.
2. ****Viewers****
   * Subscriptions waste money when idle.
   * Charge-backs possible for 120 days.
3. ****Platforms****
   * PayPal micropayments fee = 30 ¢ + 5 % → impossible below $1.
   * No public audit trail → ad-fraud ($81 B yr).

****Vilokanam-View = compulsory, streaming, sub-cent, gas-smooth, yield-bearing.****

## **D. Product Principles**

* ****Web2 UX****: one-click, no seed phrases shown.
* ****Web3 Back-end****: non-custodial, on-chain accounting, XCM yield.
* ****1-Second Granularity****: pay exactly while you watch.
* ****60-Second Float****: platform never holds >1 min of value.
* ****Closed Source Option****: no licence granted; commercial use prohibited without written permission.

## **E. Glossary**

**Table**

| **Term** | **Meaning** |
| --- | --- |
| Tick | 1-second payment unit (default 0.0001 DOT). |
| Epoch | 60-second withdrawal window. |
| Float | Viewer funds locked ≤ 60 s. |
| vDOT | DOT staking derivative received by creator. |
| OCW | Off-chain worker micro-service. |
| Vilokanam-View SDK | npm package for Web2 devs (Vilokanam-View-js). |

## **F. System Overview**

┌------------------┐

│ Web2 Front-end │ Next.js, WebRTC, Vilokanam-View-js

└------┬-----------┘

│ HTTPS / WSS

┌------┴-----------┐

│ OCW Ticker │ Rust axum, listens to RTP clock

└------┬-----------┘

│ unsigned inherent

┌------┴-----------┐

│ Vilokanam-View │ Parachain Runtime (pallets below)

│ Parachain │

└------┬-----------┘

│ XCM

┌------┴-----------┐

│ Asset-hub │ DOT staking, yields vDOT

└------------------┘

## **G. Component Deep Dive**

### **G1. Parachain Runtime**

****Tech****: Substrate 1.0, Rust nightly-2024-06-01, 2 s block time (Rococo).  
****Pallets****:

**Table**

| **Pallet File** | **Purpose** | **Storage Maps** | **Extrinsics** | **Events** |
| --- | --- | --- | --- | --- |
| pallet\_tick\_stream | Core accounting | Streams, Balances, Allowances | create\_stream, join\_stream, tick, withdraw, leave | StreamCreated, TickProcessed, Withdrawn |
| pallet\_session\_nft | Soul-bound trophy | SessionNFTs | mint\_nft | NFTMinted |
| pallet\_xcm\_staker | Float yield | StakedFloat, VDotBalance | stake\_float, unstake\_and\_send | Staked, YieldPaid |
| pallet\_verifier\_dao | Hoax protection (future) | Verifiers, Votes | verify\_alarm | Verified |

****Config Traits****:

* pallet\_timestamp for second counter.
* pallet\_balances for DOT reserve.
* orml\_xtokens for XCM transfers.

### **G2. OCW Ticker Micro-Service**

****Repo****: /ocw-ticker  
****Language****: Rust 1.78, Tokio, Axum, webrtc-rs.  
****Duties****:

1. Subscribe to block import websocket.
2. Maintain hash-map ActiveStreams.
3. For each WebRTC track, count RTP timestamp rolls (90 kHz) → 1 s boundary.
4. Push unsigned inherent tick(stream\_id, viewer, creator, 1) signed by OCW ed25519 key.
5. Every 60 ticks batch withdraw\_and\_restake.  
   ****Security****: Ticks filtered by ValidateUnsigned; payload must include Merkle root of last 90 kb WebRTC payload (proves stream is real).

### **G3. Web2 Front-End**

****Repo****: /frontend  
****Framework****: Next.js 14, React 18, TypeScript, Tailwind CSS.  
****Packages****:

* Vilokanam-View-js (private npm) wraps polkadot-js calls.
* simple-peer for WebRTC.  
  ****Pages****:
* / – landing & wallet connect.
* /create – set price, go live.
* /watch/[id] – viewer player + balance counter.  
  ****Components****:
* <PayCounter/> – real-time USD/DOT ticker.
* <WithdrawButton/> – one-click claim.  
  ****UX Flows****:
* One-time proxy wallet creation (burner key) stored in IndexedDB; gas relayed by NGNG.
* Fallback to fiat on-ramp (Stripe → BTC → DOT) if wallet empty.

### **G4. Vilokanam-View SDK (private)**

****Package****: Vilokanam-View-js (not published; internal use only)  
****API Surface****:

**TypeScript**

Copy

import { Vilokanam-View } from 'Vilokanam-View-js';

const v = await Vilokanam-View.create({ relayEndpoint: 'wss://relay.Vilokanam-View.io' });

const stream = await v.createStream({ pricePerSecond: 100000000n }); *// Planck*

await v.joinStream(stream.id, { maxSeconds: 3600 });

v.on('balanceChange', (bal) => console.log(bal));

await v.withdraw();

### **G5. XCM Float Staking**

****Flow****:

1. withdraw\_and\_restake transfers 60-second float to Asset-hub.
2. Calls staking.bondExtra.
3. Mints vDOT (via Bifrost standard) back to creator account.
4. On creator withdraw\_all burns vDOT, claims stake + yield, transfers via XCM back to parachain

****Yield Calculation****:  
yield = float \* staking\_apy \* (1/365/24/60) ≈ 0.00025 DOT per 60 s per 100 DOT float @ 14 % APY.

## **H. Token Economics**

**Table**

| **Actor** | **In** | **Out** | **Risk** |
| --- | --- | --- | --- |
| Viewer | locks DOT | content | lose seconds watched |
| Creator | content | DOT + yield | none (non-custodial) |
| OCW Relayer (NGNG) | tx fees | reimbursed by pallet treasury | spam → slashed |
| Parachain Treasury | 20 % of yield spread | pay OCW fees | configurable |
| Stakers (DOT) | security | inflation | normal staking risk |

****NGNG (No-Gas-No-Gen) Relayer****: whitelisted via pallet\_proxy, only allowed to submit tick & withdraw, refunded in block on\_finalize.

## **I. Security Analysis**

**Table**

| **Attack** | **Mitigation** |
| --- | --- |
| Fake ticks | Merkle proof of WebRTC payload + unsigned inherent filter |
| Viewer double-spend | Balances locked on-chain |
| Creator never withdraws | Auto-stake → still earns yield |
| OCW DDoS | Rate-limit inherent pool, max 10 k ticks/block |
| WebRTC IP leak | Optional TOR bridge, IP never on-chain |
| Runtime bug | Internal audits only; no external OSS review granted |

## **J. Benchmarks & KPIs**

* ****Block time****: 2 s (Rococo) → 1 s tick granularity achieved.
* ****Throughput****: 10 k ticks / block ≈ 5 M ticks/day.
* ****Tx fee****: 0.0003 DOT per 60-second batch → $0.002 per viewer-hour.
* ****Yield****: 14 % APY → creator earns ****+0.25 c/hour**** on every $100 float (vs 0 % on Twitch).
* ****Latency****: WebRTC < 300 ms; payment visual feedback < 1 s.

## **K. Development Roadmap (6-Week Sprint)**

**Table**

| **Week** | **Milestone** | **Owner** | **Done?** |
| --- | --- | --- | --- |
| 0 | Team stand-up, assign roles, create GitHub project board | All | ☐ |
| 1 | Repo scaffold, Docker compose, Rococo slot request | Runtime lead | ☐ |
| 2 | pallet\_tick\_stream coded + unit tests | Runtime lead | ☐ |
| 3 | OCW micro-service + local WebRTC tick loop | OCW dev | ☐ |
| 4 | Front-end create/watch pages + wallet connect | Front-end | ☐ |
| 5 | XCM float staking + yield payout | Runtime lead | ☐ |
| 6 | Load test 1 k viewers, shoot demo video, docs | All | ☐ |

Daily stand-up template (15 min) posted in #stand-up Discord channel.

## **L. Local Development Quick-Start**

**bash**

*# 1. Clone*

git clone https://github.com/Vilokanam-View/Vilokanam-View.git && cd Vilokanam-View

*# 2. Spin up everything*

docker compose up --build

*# 3. Front-end*

open http://localhost:3000

*# 4. Polkadot-js Apps*

open http://localhost:9944

*# 5. Run tests*

cargo test --release --all

****Folder Map****:

Copy

Vilokanam-View/

├── node/ # Substrate node

├── pallets/ # All pallets

├── ocw-ticker/ # Rust micro-service

├── frontend/ # Next.js

├── sdk-js/ # Vilokanam-View-js (private)

├── docs/ # This file, audits, ADRs

└── scripts/ # CI, deploy, benchmark

## **M. Deployment & CI/CD**

* ****GitHub Actions****:
  + rust.yml – cargo test, clippy, fmt.
  + frontend.yml – lint, build, push to Vercel.
  + docker.yml – multi-arch build, push to GHCR (private).
* ****Environments****:
  + dev → auto deploy on main push.
  + staging → manual trigger, Rococo.
  + prod – slot winner, Kusama → Polkadot.

## **N. API Reference (Internal Only)**

Open https://docs.Vilokanam-View.io/rpc (private redoc) after docker compose up.  
Key RPCs:

* Vilokanam-View\_createStream(pricePerSecond)
* Vilokanam-View\_joinStream(streamId, maxSeconds)
* Vilokanam-View\_withdraw(streamId)

## **O. Regulatory & Compliance**

* No custody > 60 s → ****not a money transmitter**** in most jurisdictions.
* KYC/AML responsibility lies on fiat on-ramp provider (Stripe).
* GDPR: No IP or personal data stored on-chain; optional consent checkbox.

## **P. Marketing & Go-To-Market**

****Phase 0 (Hackathon)****

* Tweet 30-sec clip of counter spinning ⏱️💸.
* Tag #PolkadotHackathon #Vilokanam-View.

****Phase 1 (0-3 mo)****

* Integrate with OBS plugin (C++).
* Partner with small Twitch streamers (10–100 viewers) → instant payout hook.

****Phase 2 (3-12 mo)****

* Zoom SDK app → paid webinars.
* Substack “paid podcast” button.

****KPI****

* 1 k creator sign-ups → 100 k viewer-hours → $7 k total flow → $1.4 yield paid.

## **Q. Competitive Landscape**

**Table**

| **Product** | **Granularity** | **Custody** | **Yield** | **Blockchain** |
| --- | --- | --- | --- | --- |
| Twitch | 1 month | full | 0 % | ❌ |
| YouTube SuperChat | voluntary | full | 0 % | ❌ |
| Coil (Interledger) | 1 second | full | 0 % | ❌ |
| SatoshiStream (BTC) | voluntary | self | 0 % | BTC |
| ****Vilokanam-View**** | ****1 second**** | **≤**60 s**** | ****14 % APY**** | ****Polkadot**** |

## **R. Known Limitations**

* Requires DOT liquidity (viewer must have DOT or fiat on-ramp).
* WebRTC packet proof adds ~300 bytes/tick → 3 MB/block at 10 k ticks (still < 50 % block weight).
* Float staking subject to Polkadot unbonding 28 days (creators hold vDOT liquid).

## **S. Future Roadmap (Post-Hack)**

* Audio-only & screen-share mode.
* Mobile React-Native SDK.
* Fiat on-ramp Stripe → DOT (auto-swap).
* Advertiser auction: brands bid to cover viewer cost for 30 s ad slot.
* AI-generated captions NFT minted per session.

## **T. Contribution Policy**

****Closed source; no external contributions accepted without NDA and written agreement.****  
For commercial licensing contact: [builders@Vilokanam-View.io](mailto:builders@vilokanam.io)

## **U. Contact**

* Twitter: [@Vilokanam-View](https://twitter.com/vilokanam)
* Discord: [https://discord.gg/Vilokanam-View](https://discord.gg/vilokanam) (invite-only)
* Email: [builders@Vilokanam-View.io](mailto:builders@vilokanam.io)

****© 2025 Vilokanam-View – All Rights Reserved.****