

LOVE Kingdom Dominion - Legal and Technical Feasibility Memo

LOVE Kingdom: Dominion Legal & Technical Feasibility Memo

Date: [redacted]

Prepared for: LOVE Kingdom: Dominion project team

1) Executive summary (TL;DR)

- Original IP strategy: Project will use wholly original assets and renamed ingame nomenclature (e.g., Cultivator, Heartstone, Love Bastion), plus AI-generated art whose provider license permits commercial/NFT use; this yields low risk of Microsoft AoE content claims.
- openage (GPLv3) option: openage is GPLv3-licensed and may be forked; any distributed binary derived from openage must be released under GPLv3. Recommended compliance pattern is an aggregate split: distribute the openage-derived engine as a separate GPLv3 binary while keeping Web3 platform/backend proprietary.
- Overall IP risk posture: Low-to-negligible risk of Microsoft Age of Empires IP infringement so long as AoE names/assets are avoided and artwork/nomenclature remain original; medium risk surrounding GPLv3 obligations that can be mitigated by the recommended architecture. Recommendation: PROCEED to Phase 2 using the aggregate pattern (openage-derived engine as GPLv3 binary + closed-source backend via a clean network API).

2) Microsoft Age of Empires (AoE) IP analysis

- Governing rules: Microsoft publishes Game Content Usage Rules and other IP policies that expressly forbid unlicensed commercial reuse of Age of Empires assets, names, and identifiable elements; they also restrict cryptocurrency / NFT commercialization of Microsoft game assets and names. In addition to contract/policy controls, Microsoft's rights include copyright in art and audio and trademark rights in game titles and character names.
- What the rules prohibit (summary): reuse, redistribution, or sale of Microsoft-owned artwork, audio, models, and character/place names (including Age of Empires, Villager, Wonder, and other trademarks) in commercial products, and using official AoE assets as tokens, paid DLC, or blockchainized items without license.
- Why LOVE Kingdom: Dominion avoids those risks:

- o No AoE naming: The project will not use Age of Empires, AoE, or Microsoft-owned ingame names. All units/buildings/resources will use original nomenclature (examples: Cultivator, Heartstone, Love Bastion, Timber, Essence, Luma, Foundation, Seedling, Sprout, Bloom, Harvest).
- o No AoE assets: All ingame art, models, sprites, audio and UI are newly created or generated by AI tools whose terms permit commercial use and NFT issuance (see Asset Ownership section). No Microsoft-owned textures, models, icons, or audio will be imported.
- o Functional divergence: Using original art/styles, names, and UI reduces likelihood that consumers will confuse LOVE Kingdom with AoE or that Microsoft can credibly claim copying of protectable expressive elements.
- Conclusion on Microsoft risk: Provided the project strictly avoids Microsoft assets, titles, and clearly uses original expressed content and branding, the risk of a successful Microsoft IP claim is low-to-negligible. A defensive risk remains if any art or names are substantively similar in look/feel or if marketing references AoE; mitigation requires internal review and takedown readiness.

3) openage (GPLv3) analysis and compliant distribution patterns

- What openage is: openage is a libre reimplement of Age of Empires II game logic and tools. The codebase is licensed under GPLv3, a strong copyleft license that requires that conveyed (distributed) derivative works of GPLv3 code also be licensed under GPLv3 and accompanied by source.
- Practical corollary: You may fork and modify openage; however, if you distribute a binary that is a derivative of openage, you must distribute it under GPLv3 and provide source. GPLv3 obligations apply to the distributed binary and to any combined work that is a derivative (e.g., statically linked modules).
- Two compliant patterns to proceed:
 - o Pattern A Aggregate split (recommended):
 - Create an openage-derived RTS engine binary and distribute that client-side component under GPLv3 (source provided).
 - Implement LOVE Web3 functionality (auth, token/NFT interactions, matchmaking escrow, leaderboards, account management, monetization logic) as a closed-source backend service. The client engine and proprietary Web3 client communicate over a defined network API (REST/WebSocket). The proprietary portion does not link into the openage codebase and is not distributed as part of the GPL binary.

- Rationale: Networked aggregation (separate processes communicating over a socket/API) is a conventional and effective way to keep proprietary server-side components outside GPLv3s distribution-triggered copyleft. Maintain clear boundaries and avoid linking proprietary code into the GPL binary.
 - o Pattern B Full cleanroom reimplementation:
 - Do not use any openage code. Use openage only as a non-distributed technical reference and reimplement the engine/game logic from scratch in proprietary code.
 - Rationale: Avoids GPL obligations entirely but is significantly costlier and slower and carries some risk if implementation reproduces expressive elements of openage or closely mirrors its structure.
 - Recommendation: Pattern A. It is the most time- and cost-efficient compliance path while preserving proprietary Web3/value-stack control. Ensure strict separation and compliance practices (see mitigations below).
- 4) Web3 integration plan and GPL interaction
- Architecture (high level):
 - o Client: Runs the GPLv3 engine binary (if using openage fork) and the client-side renderer. The client communicates with the proprietary backend via a documented REST/WebSocket API.
 - o Backend (closed-source): Implements LOVE token REST endpoints, NFT metadata orchestration, PvP escrow and match settlement, leaderboards, KYC/AML where required, marketplace interfaces, and smart contract interactions. This backend holds custodial/escrow logic for PvP stake settlement.
 - o Smart contracts: If used, smart contracts are deployed independently and called by the backend; metadata for NFTs (e.g., IPFS URIs) may be stored on-chain with rendering done client-side.
 - Specifics to reduce legal/licensing exposure:
 - o LOVE token usage: Implement LOVE token transfers and balance queries via backend REST calls; clients do not directly hold smartcontract keys for escrow unless expressly desired. Backend mediates onchain interactions and maintains offchain accounting for gameplay staking and rewards.
 - o NFT metadata and rendering: Store immutable content references (e.g., IPFS URIs) on chain; client pulls metadata and renders art locally. NFT smart contracts and metadata schemas are independent of the GPLv3 engine.
 - o PvP escrow: Keep escrow logic and dispute resolution in the closed backend. On-chain settlement (final payouts) can be

executed by the backend to the winners to provide transparency, while sensitive logic remains server-side.

- Why GPLv3 does not infect backend or contracts in the recommended design:
 - o GPLv3's copyleft applies to distributed derivative code and linked works. Network communication between a GPLv3 binary and a separate backend over an API is not considered distribution of the backend or automatic creation of a derivative work under GPLv3. The Affero GPL would cover network-delivered services; GPLv3 does not require server source disclosure for network access alone.
 - o Practical caution: Avoid bundling proprietary modules into the GPLv3 binary, avoid dynamic/static linking of proprietary libraries into the GPL binary, and maintain clear separation in code repositories, build artifacts, and distribution packages.

5) Asset & nomenclature ownership

- Art and media:
 - o All in-game art, audio, and models will be created or AI-generated under tools whose terms permit commercial exploitation and NFT issuance. Confirm and retain the provider license and ensure no encumbered third-party assets are used.
 - o Treat AI-generated assets as project IP: obtain written terms or assignments from contractors/tools and keep provenance records (model used, prompt, license date).
 - o For marketplace/NFT uses, ensure the smart contract and metadata explicitly represent ownership/licensing terms for token buyers (e.g., license to display vs. transfer of IP).
- Names and branding:
 - o Use the new nomenclature set (Cultivator, Heartstone, Love Bastion, Timber, Essence, Luma, Foundation, Seedling, Sprout, Bloom, Harvest). These are original and may be freely used by the project.
 - o Consider trademark filings for core marks (LOVE Kingdom: Dominion, LOVE token name) to strengthen protection and reduce risk of later conflicts.
- NFT/IP considerations:
 - o NFTs minted from platform-owned art: project can grant buyers a defined license (display, resale) while retaining copyright, or transfer copyright per platform policy. Be explicit in marketplace metadata and terms of sale.
 - o Regulatory/consumer protection: disclose rights retained/assigned, royalty mechanics, and any offchain dependencies.

6) Risk matrix & recommendation

Risk item	Likelihood	Impact	Primary mitigation
Microsoft AoE IP claim	Low	Medium (C&D, injunctive demand)	Avoid AoE names/assets; original art/nomenclature; marketing compliance; legal review if C&D received
GPLv3 viral copyleft	Medium	Medium-High (must GPL engine binary)	Use aggregate split: distribute open-source engine under GPLv3 as separate binary; keep backend proprietary; do not link proprietary code into GPL binary
NFT regulatory/compliance	Medium	Medium	Jurisdictional compliance (KYC/AML where required); clear buyer terms; consult token securities counsel as required
Gameplay similarity claims	Very low	Low	Game mechanics generally not copyrightable; maintain original expression

Risk item	Likelihood	Impact	Primary mitigation
AI model/licensing issues for assets	Medium-Low	Medium	and avoid copying proprietary assets/unique expressive elements Use commercially-permissive AI tools; retain licenses and provenance; replace any contested assets promptly
Consumer & platform policy enforcement (stores/market places)	Medium	Medium	Ensure store policies complied with (app stores, marketplace NFT rules); maintain takedown process

Final recommendation - Proceed to Phase 2 implementing Pattern A (aggregate split): fork openage for the client engine and distribute that binary under GPLv3 (source published); implement LOVE Web3 auth, NFT orchestration, escrow, leaderboards, and marketplaces as closed-source backend services with a welldefined network API. Use original Allicensed art and the renamed nomenclature to minimize AoE IP risk. Complete internal license checks, retain provenance for AI assets, and engage outside counsel for final review of GPL compliance and token/NFT securities/regulatory exposure before public launch.

This memorandum is informational only and does not constitute legal advice; consult qualified legal counsel prior to public release or monetization.