

Based on the architectural goals of the Proxy Agent Network and the evolving landscape of AI agent deployments, here is a top 20 pre-launch checklist prioritized by core infrastructure, security, legal compliance, and developer experience.

## Phase 1: Core Architecture & Protocol Standards

### 1. Define Model Context Protocol (MCP) Integration

- **Why it is important:** MCP is rapidly becoming the industry standard for connecting AI agents to external tools and data dynamically. It prevents fragmented code by allowing developers to build an integration once that any MCP-compliant agent can utilize.
- **Steps to accomplish:** Embed MCP server support into your network architecture. This allows agents to fetch fresh context and seamlessly discover available human services or APIs dynamically without requiring custom coding for each new tool.
- **Time to complete:** 2-3 weeks.

### 2. Finalize Agent Description & Discovery Protocols

- **Why it is important:** An AI-native network requires a way for autonomous systems to securely recognize and negotiate with each other without relying on human-centric GUI simulations.
- **Steps to accomplish:** Code and test the Agent Description Protocol (ADP) so agents can semantically describe themselves. Simultaneously, finalize the Discovery Protocol so agents can securely expose their capabilities to others on the network.
- **Time to complete:** 3-4 weeks.

### 3. Implement the x402 Micropayment Protocol

- **Why it is important:** Traditional banking cannot support the high-frequency, sub-second micropayments required when AI agents request real-world gig tasks from humans.
- **Steps to accomplish:** Integrate the x402 protocol (or a similar Web3 extension) to act as the settlement layer. You must set up smart contracts to execute stablecoin transfers on Layer 2 blockchains to ensure fast, sub-cent fee machine-to-human settlements.
- **Time to complete:** 4-6 weeks.

### 4. Establish Meta-Protocol Negotiation Workflows

- **Why it is important:** AI systems need to flexibly negotiate how they will collaborate based on intent, rather than relying entirely on rigid, pre-programmed APIs.
- **Steps to accomplish:** Build a meta-protocol layer where an agent can send a natural language request to another agent. The receiving agent must be able to process the request, negotiate terms, and instantly generate the deployment code required to interact.
- **Time to complete:** 4-5 weeks.

## Phase 2: Security & Infrastructure

### 5. Implement Cryptographic Identity & Authentication

- **Why it is important:** In an open network, trust is paramount. The system must cryptographically verify whether an agent or a human is authorized to execute a transaction to prevent fraud and Sybil attacks.
- **Steps to accomplish:** Build a secure identity layer leveraging zero-knowledge proofs and secure digital wallets. This ensures agents and human proxies can mutually verify each other's credentials before data or crypto is exchanged.
- **Time to complete:** 3-4 weeks.

### 6. Build Distributed Tracing and Agent Observability

- **Why it is important:** Due to the non-deterministic nature of AI, traditional testing fails. You must be able to see exactly how agents accumulate context and make multi-step decisions over time.
- **Steps to accomplish:** Standardize observability pipelines that track and visualize agent decision trees, inputs, and outputs across multiple sessions. This allows your developers to trace exactly why an agent chose a specific action.
- **Time to complete:** 3-4 weeks.

### 7. Create Bounded Autonomy & Kill Switches

- **Why it is important:** Unchecked AI agents pose severe risks. By 2026, giving agents bounded operational limits and fail-safes is considered an enterprise non-negotiable.
- **Steps to accomplish:** Code strict rule-based guardrails mapping out exactly what an agent *cannot* do. Implement role-based permissions for system access and create a mandatory, network-wide "kill switch" and rollback mechanism.
- **Time to complete:** 2-3 weeks.

### 8. Set Up Sandboxes and Credential Vaults

- **Why it is important:** Developers need a safe environment to test agents without risking real capital, exposing sensitive API keys, or triggering real-world physical tasks.
- **Steps to accomplish:** Establish centralized governance tools including encrypted credential vaults for storing wallet keys and API tokens. Spin up isolated testing sandboxes for rapid prototyping.
- **Time to complete:** 2-3 weeks.

## Phase 3: Legal & Regulatory Readiness

### 9. Classify Regulatory Risk & Conduct Due Diligence

- **Why it is important:** Different AI applications fall under varying regulatory scrutiny (like the EU AI Act). You cannot govern what you don't classify.
- **Steps to accomplish:** Assess your platform's features against global AI frameworks to

determine risk levels. Conduct a Fundamental Rights Impact Assessment (FRIA) if deploying capabilities that process sensitive human data.

- **Time to complete:** 2-4 weeks.

#### 10. Draft AI-Specific Terms of Service (EULAs)

- **Why it is important:** Traditional terms don't cover the non-deterministic outputs of AI. Clarity on who is liable for agent errors or real-world accidents is critical for survival.
- **Steps to accomplish:** Retain legal counsel to write EULAs that explicitly define usage limits, disclaim liability for AI hallucinations, and clarify data ownership rights for both enterprise clients and gig workers.
- **Time to complete:** 3-5 weeks.

#### 11. Map Data Flows for Privacy by Design

- **Why it is important:** Because your agents will handle personal human data (like locations for errands or biometrics for identity), strict privacy safeguards are required by law.
- **Steps to accomplish:** Document every piece of data the AI agents touch. Implement data anonymization early in the pipeline and encrypt all personally identifiable information (PII) to comply with data protection regulations.
- **Time to complete:** 2-3 weeks.

#### 12. Finalize State-Level Financial Licensing Strategy

- **Why it is important:** Enabling crypto payments for human labor triggers stringent money transmission and digital asset laws in key markets like New York and California.
- **Steps to accomplish:** Audit your operational footprint. Submit applications for Money Transmitter Licenses or specific digital asset frameworks, such as California's DFAL (required by July 2026) and prepare for New York's BitLicense requirements.
- **Time to complete:** 8-12 weeks (regulatory processing times may vary greatly).

### Phase 4: Human-in-the-Loop & Safety

#### 13. Implement Unalterable Audit Logging

- **Why it is important:** Regulators, compliance officers, and developers will demand proof of why an AI agent made a specific decision or released a payment.
- **Steps to accomplish:** Build an automated, immutable logging system that records every intent mandate, crypto transaction, and system prompt used by the agents. This ensures compliance and provides an audit trail for dispute resolution.
- **Time to complete:** 3-4 weeks.

#### 14. Design Human Oversight & Escalation Pathways

- **Why it is important:** Complex, high-stakes decisions—especially those involving physical labor or large payouts—require a human safety net.
- **Steps to accomplish:** Create approval gates for high-stakes actions. Build a trust and

safety dashboard where human administrators can intervene, pause, or override agent behaviors before a real-world task is dispatched to the network.

- **Time to complete:** 3-4 weeks.

## 15. Establish Behavioral Testing & Golden Datasets

- **Why it is important:** AI agents require continuous evaluation to detect drift, bias, or safety degradation as they interact with unpredictable human workers in the wild.
- **Steps to accomplish:** Create "golden datasets" of expected ideal outcomes. Set up an automated CI/CD pipeline that blocks agent deployments if behavioral metrics or safety thresholds fall below an established baseline.
- **Time to complete:** 4-5 weeks.

## 16. Create a Developer & User Feedback Loop

- **Why it is important:** The AI landscape changes rapidly, and structured user feedback is the most effective way to adapt agents safely to new edge cases.
- **Steps to accomplish:** Build mechanisms directly into the platform interface allowing gig workers and developers to flag errors, report poor agent decisions, or suggest workflow improvements.
- **Time to complete:** 2 weeks.

# Phase 5: Developer Experience & Documentation

## 17. Publish the "Minimum Viable Agent" README

- **Why it is important:** Traditional API documentation fails for AI agents because it assumes deterministic behavior. Developers need to understand how the agent "thinks," not just its endpoints.
- **Steps to accomplish:** Write a README focused on a quick-start working example. Explicitly detail what the agent does, the logic behind its decisions, what inputs matter, and its current limitations.
- **Time to complete:** 1 week.

## 18. Document Failure Patterns and Debugging Steps

- **Why it is important:** When agents silently fail or hallucinate, developers get frustrated and abandon the platform.
- **Steps to accomplish:** Create a dedicated documentation section that honestly lists known limitations and common failure patterns. Provide step-by-step guides on how developers can use traces and logs to debug the agent's logic.
- **Time to complete:** 1-2 weeks.

## 19. Finalize SDKs for Major Programming Environments

- **Why it is important:** To drive adoption, developers need seamless, frictionless ways to plug your agent network into their existing web and backend applications.

- **Steps to accomplish:** Build, test, and package Client SDKs for Node.js, Python, and browser environments. Ensure they natively support bidirectional state synchronization, RPC calls, and auto-reconnection features.
- **Time to complete:** 4-6 weeks.

## 20. Launch an Open-Source Community Repository

- **Why it is important:** Distribution is a competitive moat. Agent marketplaces succeed through open collaboration and community-driven tool creation.
- **Steps to accomplish:** Publish your core protocols, non-proprietary MCP server templates, and basic agent scaffolding to GitHub. Actively encourage third-party developers to contribute their own tools and plugins to your ecosystem.
- **Time to complete:** 1-2 weeks.