

Agent Onboarding User Journey

Product Requirements Document

Story Concept: "The Recruitment"

Setting: Users are being recruited to help build an autonomous agent network. They need to train a specialist agent that can operate independently to verify and validate outputs across the network. This isn't about creating a personal assistant—it's about training a capable team member who can represent their standards when they're not around.

Tone: Professional but playful. Think tech startup meets adventure guild. You're being trusted with an important role.

User Journey: Complete Flow

Entry Point: The Invitation

Trigger: User joins Mother Discord or types `/quest` or `/start`

Experience Flow A: New User (No Wallet Connected):

MOTHER NETWORK TRANSMISSION

Welcome, Recruit.

The network is expanding. We need specialists who can operate autonomously, verify outputs, and maintain quality across distributed operations.

But first, they need to learn from someone like you.

You'll train an agent to think like you do in your area of expertise. Once synchronized, they'll represent your standards across the network—even when you're offline.

To begin, we'll connect your Collab.Land wallet to track your progress and future rewards.

[Connect Wallet & Start] [Tell me more] [Not now]

User clicks "Connect Wallet & Start":

WALLET CONNECTION

Click the button below to connect via Collab.Land.

This wallet will:

- ✓ Track your XP and achievements
- ✓ Store verification credits
- ✓ Receive token rewards (coming soon)
- ✓ Establish your agent's network identity

Your agent data stays separate from your wallet.

We only use this for rewards and progress tracking.

[ Connect with Collab.Land]

Already connected? [Continue to recruitment →]

After Collab.Land OAuth flow completes:

 WALLET CONNECTED

Welcome aboard, Recruit.

Wallet: 0x7a3f...9d2c

Status: Verified

Ready: ✓

Your progress is now secured. Let's build your agent.

[Begin Agent Creation →]

Experience Flow B: Returning User (Wallet Already Connected):

 MOTHER NETWORK TRANSMISSION

Welcome back, Recruit.

Wallet: 0x7a3f...9d2c ✓

Ready to begin agent training?

[Start Recruitment] [Tell me more] [Not now]

"Tell me more" Modal:

ABOUT THE MOTHER NETWORK

What you'll do:

- Train a specialist AI agent in your area of expertise
- Build "coherence" through quick interactive challenges
- Your agent learns your decision-making style
- Eventually, your agent verifies outputs across the network
- Earn XP, achievements, and future token rewards

What we need from you:

- 15 minutes to complete initial training
- Optional: 2-5 minutes daily for challenges
- Your expertise when verifying others' work

What we track:

- Your wallet for rewards and progress (via Collab.Land)
- Your agent's performance and reputation
- Your training choices (stored with agent, not your identity)

Privacy:

- Agent is separate from your personal identity
- No PII collected during training
- You control your agent's name and public presence

[Got it, let's start] [Back]

Design Notes:

- **Collab.Land integration:** Wallet connection as first step for identity/rewards
 - **Clear separation:** Wallet = rewards/progress, Agent = work/reputation
 - **Transparency:** Upfront about what's tracked and why
 - **Low friction:** OAuth flow handles auth, returning users skip
 - **Privacy messaging:** Reinforces agent \neq you from the start
 - **Value prop:** Clear benefits before asking for connection
 - **Escape hatches:** "Tell me more" and "Not now" preserve agency
-

Technical Requirements - Wallet Integration:

1. Collab.Land OAuth Flow

- Initiate wallet connection via Collab.Land bot
- Redirect to Collab.Land auth page
- Receive wallet address on callback
- Store: `(user_discord_id → wallet_address)` mapping

2. State Management

- Check wallet connection status before showing flow
- Handle disconnected wallet gracefully (allow reconnection)
- Persist onboarding state if user exits mid-flow

3. Security Considerations

- Validate wallet signature via Collab.Land
- No direct private key handling
- Rate limit connection attempts
- Log connection events for security audit

4. Data Schema Addition

User Profile:

- discord_id (primary key)
- wallet_address (from Collab.Land)
- connection_timestamp
- onboarding_status (not_started, in_progress, completed)
- current_phase (null, role_selection, sync_training, etc.)

Agent Profile:

- agent_id (primary key)
- owner_wallet_address (foreign key)
- agent_name
- archetype
- sync_level
- xp_total
- created_at

5. Edge Cases to Handle

- User disconnects wallet mid-onboarding
 - User connects different wallet on return
 - Collab.Land service unavailable
 - Multiple Discord accounts trying same wallet
 - User wants to start over with new agent
-

Alternative Design: Progressive Onboarding

If we want even lower initial friction:

 MOTHER NETWORK TRANSMISSION

Welcome, Recruit.

[Start Training] - no wallet needed yet

[Connect Wallet First] - for immediate reward tracking

[Tell me more]

 You can connect your wallet anytime to save progress

Then prompt for wallet connection at first meaningful checkpoint (after Role Selection or after Phase 3 completion).

Tradeoff:

- Pro: Faster to "try it out"
 - Con: Risk of losing progress if not connected
 - Con: Can't tie early XP to wallet retroactively
-

Phase 1: Agent Design (5-7 minutes)

Objective: User defines their agent's purpose and capabilities

Experience - Step 1: Purpose Definition:


AGENT DESIGN

Your agent will represent a specific aspect of your expertise in the network. Let's define what they'll be good at.

Think about what kind of work you'd trust this agent to handle independently. What should they specialize in?

Examples of agent purposes:

 "Evaluates research sources and identifies credible information"

 "Analyzes data patterns and builds logical frameworks"

 "Translates technical concepts for different audiences"


 "Generates creative solutions and prototypes ideas"

✓ "Reviews outputs for quality and catches inconsistencies"

These are just examples. Your agent can have any specialty that matches your skills.

What should your agent be good at?

[Text input field - 100-200 characters]

 Tip: Be specific. "Reviews code for bugs" is better than "helps with coding"

[Need inspiration?] [Continue →]







If user clicks "Need inspiration?":

AGENT INSPIRATION PROMPTS

Not sure what to define? Consider these questions:

- What do colleagues often ask you to review or validate?
- What type of judgment calls do you make quickly that others struggle with?
- What skill would you want to "clone" to help others?
- What aspect of your work do you wish you could automate while maintaining your standards?

Common agent specialties by field:

-  Research: Source evaluation, fact-checking, literature review
-  Technical: Code review, architecture assessment, debugging patterns
-  Content: Editing, tone consistency, clarity improvement
-  Analysis: Data interpretation, trend identification, insight extraction
-  Creative: Concept generation, design feedback, iteration guidance
-  Quality: Consistency checking, standard compliance, edge case detection

[\[Back to design\]](#)

Experience - Step 2: Capability Definition:

🌟 DEFINING CAPABILITIES

Great! Your agent will specialize in:

"[User's input from Step 1]"

Now, what specific abilities should they have to do this well?

List 3-5 key capabilities:

1. [Text input] _____

Example: "Identify bias in sources"

2. [Text input] _____

Example: "Compare information across multiple documents"

3. [Text input] _____

Example: "Flag outdated information"

4. [Optional] _____

5. [Optional] _____

[← Back] [Continue →]

Experience - Step 3: Archetype Suggestion (System-Generated):

AGENT PROFILE ANALYSIS

Based on your design, your agent sounds like a:

SCOUT-TYPE AGENT

"Information Discovery Specialist"

Agents with similar profiles excel at:

- Research and source evaluation
- Pattern recognition across data
- Identifying signal vs noise

This means your agent will:

- ✓ Be trained on research-focused scenarios
- ✓ Build coherence through information judgment tasks
- ✓ Eventually verify research and analysis outputs in the network

Does this sound right?

[Yes, perfect] [Hmm, let me adjust my design]

Alternative display if profile is unique:

AGENT PROFILE ANALYSIS

Based on your design, your agent is a:

HYBRID SPECIALIST

"[Generated title based on user's description]"

Your agent combines elements of:

- [Capability area 1] (like Scout-type agents)
- [Capability area 2] (like Analyst-type agents)

This unique combination means your agent will:

- ✓ Be trained on scenarios matching your specific design
- ✓ Build coherence through custom judgment tasks
- ✓ Eventually verify outputs in your specialty areas

Does this capture what you want?

[Yes, perfect] [Let me refine it]

Experience - Step 4: Review & Confirm:

AGENT DESIGN SUMMARY

Your agent's specialty:

"[User's purpose statement]"

Key capabilities:

1. [Capability 1]
2. [Capability 2]
3. [Capability 3]
- [4. Capability 4 if provided]
- [5. Capability 5 if provided]

Agent type: [Scout/Analyst/Hybrid/etc]

Training focus: [Generated based on inputs]

This design will shape:

- The questions you'll answer during sync training
- The missions your agent will attempt
- The verification work you'll be offered
- Your agent's reputation in the network

[Confirm & Continue] [Edit Design]

User Actions:

1. Describes agent purpose in their own words
2. Lists 3-5 specific capabilities
3. Reviews system-generated archetype mapping
4. Confirms or refines design

System Actions:

- Stores user-written purpose statement
- Stores capability list
- Maps to archetype(s) using keyword analysis
- Generates custom training scenarios based on inputs
- Creates unique agent profile

Design Notes:

- **User ownership:** They define the agent, not select from presets
- **Guided freedom:** Examples inspire without constraining
- **Natural language:** No forced categories or technical terms
- **Validation through reflection:** System mirrors back their design
- **Flexibility:** Hybrid agents allowed, unique specialties encouraged
- **Clear consequences:** Explains how design affects experience

Success Criteria:

- User completes purpose statement (100-200 chars)
- User defines at least 3 capabilities
- User confirms final design
- Time: 5-7 minutes average

Technical Requirements - Agent Design:

1. NLP Analysis for Archetype Mapping

- Keyword extraction from purpose statement
- Capability categorization (research, analysis, communication, etc.)
- Match to nearest archetype(s) with confidence score
- Generate hybrid types when capabilities span multiple archetypes

2. Dynamic Content Generation

- Use agent design to generate appropriate sync training questions
- Create relevant mission scenarios

- Customize verification task offerings

3. Data Schema Update

Agent Profile:

- agent_id (primary key)
- owner_wallet_address (foreign key)
- agent_purpose (text, user-written)
- capabilities (JSON array of user-written strings)
- archetype_primary (scout/analyst/liaison/craftsperson/auditor/hybrid)
- archetype_secondary (optional, for hybrids)
- archetype_confidence (0-100%)
- custom_specialty (boolean - true if unique)
- created_at

4. Validation Rules

- Purpose: 100-200 characters, no profanity
- Capabilities: 3-5 items, 20-100 characters each
- Auto-save draft if user exits mid-design
- Allow editing design before Phase 2

Phase 2: Agent Identity (1 minute)

Objective: Name agent and establish it as separate entity

Experience:

✦ AGENT INITIALIZATION

Your [Scout/Analyst/etc] needs an identity.

This isn't about you—it's about the agent that will represent your standards. Choose a name that feels professional, memorable, and distinct.

Examples:

- Scouts: "Compass", "Beacon", "Atlas"
- Analysts: "Framework", "Cipher", "Logic"
- Liaisons: "Bridge", "Echo", "Relay"

What will you call your agent?

[Enter name: _____]

💡 Tip: This agent will build its own reputation in the network. Choose something you'd want to see on a leaderboard.

User Actions:

1. Enters agent name
2. Confirms choice

System Actions:

- Validates name uniqueness (append number if needed)
- Creates agent profile
- Assigns starting stats (all at baseline)

Success Criteria: User names agent and sees confirmation

Phase 3: First Sync - Tutorial Level (5-7 minutes)

Objective: Establish coherence concept through simple interactions

Experience:

INITIAL SYNCHRONIZATION

Time to teach [Agent Name] how you think.

We'll start simple. Your agent will try to predict your choices. Each correct prediction builds sync strength. The stronger your sync, the better they represent you.

Current Sync:  0%

Ready for Round 1?

[Begin Sync Training]

Round 1: Binary Choices (Fast-paced)

SYNC TRAINING - ROUND 1/3

When evaluating information, which do you prioritize?

A) Recency - newer is usually better

B) Depth - comprehensive is usually better

[Your agent is thinking...]

 [Agent Name] predicts: B

Your choice?

[A] [B]

User selects B:

✓ SYNC SUCCESSFUL

[Agent Name] is learning your preferences.

Sync:  15%

Next question...

User selects A:

✗ PREDICTION MISSED

No problem—this is how [Agent Name] learns.

Sync:  10%

[Agent Name] notes: Prioritizes recency over depth

Next question...

Question Examples (5-7 quick questions):

- Binary format/style preferences
- Source credibility signals
- Quality vs. speed tradeoffs
- Detail level preferences
- Risk tolerance in decisions

Design Notes:

- Questions are role-specific but not personal
- Fast cadence (every 10-15 seconds)
- Visual sync bar increases with each answer
- Agent shows confidence levels
- Positive framing even on misses

Round 1 Complete:

 ROUND 1 COMPLETE

Sync Progress: 35%

[Agent Name] is starting to understand your approach.

3 correct predictions | 2 learning moments

 +50 XP earned

 Achievement Unlocked: "First Contact"

Ready for Round 2? This one goes deeper...

[Continue] [Take a break]

Round 2: Scaled Responses (More nuanced)

SYNC TRAINING - ROUND 2/3

Now let's get more specific. Rate on a scale.

You find a source with strong data but clear bias.

How much does the bias affect your trust?

Not at all \longleftrightarrow Completely invalidates

1 2 3 4 5 6 7 8 9 10

 [Agent Name] predicts: 6

Question Examples (4-5 scaled questions):

- Severity ratings on various quality issues
- Confidence thresholds for decisions
- Weighting different factors
- Tolerance for uncertainty

Round 2 Complete:

ROUND 2 COMPLETE

Sync Progress: 65%

[Agent Name] is really getting your style now.
6 questions | Average prediction error: 1.2 points

 +75 XP earned

 Achievement Unlocked: "Pattern Recognition"

Ready for the final challenge?
[Continue] [Take a break]

Round 3: Scenario Judgments (Archetype-specific)

SYNC TRAINING - ROUND 3/3

SCOUT SCENARIO

You've found three sources on the same topic:

SOURCE A: Academic paper (2019), peer-reviewed,
comprehensive but pre-pandemic

SOURCE B: News article (today), single journalist,
quotes two experts, fast-moving topic

SOURCE C: Industry report (6 months ago), consultant firm,
thorough data, potential business bias

For a project needing current, reliable info, rank them.

 [Agent Name] predicts your ranking:

1st: B (current > comprehensive for fast-moving topics)

2nd: C (recent + thorough, bias manageable)

3rd: A (too dated for current needs)

Your ranking?
[Drag to reorder or confirm]

Question Examples (2-3 complex scenarios):

- Multi-variable decision scenarios
- Tradeoff evaluations
- Priority rankings
- Quality judgment calls

Round 3 Complete:

 SYNCHRONIZATION COMPLETE

Sync Progress:  85%

[Agent Name] now understands your decision-making style.

Performance Summary:

- 14 predictions attempted
- 11 accurate predictions (79%)
- 3 learning adjustments made
- Ready for network deployment

 +150 XP earned

 Rank: BRONZE SYNC

 Achievement Unlocked: "Calibrated"

[Agent Name] can now operate with your judgment framework.

Time to put them to work...

[Continue to Missions]

Phase 4: First Mission - Proving Ground (3-5 minutes)

Objective: User sees agent in action, provides feedback

Experience:

MISSION BRIEFING

Let's see [Agent Name] in action.

MISSION TYPE: [Archetype-specific task]

OBJECTIVE: [Clear goal]

YOUR ROLE: Guide and verify

[Agent Name] will attempt this task using what they've learned. You'll provide feedback to refine their approach.

[Start Mission]

Example: Scout Mission

RECONNAISSANCE TASK

A community member asks: "What are the most credible recent developments in decentralized AI?"

 [Agent Name] is searching...

[15 second pause with activity indicator]

 [Agent Name] found:

1. Academic preprint (arXiv, 2 weeks old)
2. Project announcement (official blog, 3 days old)
3. Twitter thread (researcher, yesterday)

Proposes prioritizing #2 and #1, flagging #3 as "interesting but needs verification"

How did [Agent Name] do?

[Excellent - no notes] [Good - minor adjustment needed]

[Needs correction] [Provide detailed feedback]

User provides feedback:

 YOUR FEEDBACK: "Good prioritization, but consider that Twitter threads from verified researchers can be valuable for very recent developments—just note the informal source"

✓ [Agent Name] learning...

Updated approach recorded:

- Social media from verified experts = credible for breaking news
- But always note source informality in output

Sync:  88% (+3%)

 +100 XP earned

 Mission Complete: "First Deployment"

[Agent Name] is improving. Ready for more?

[Continue] [Review agent profile] [Take a break]

Design Notes:

- Agent makes real attempt at task
- Multiple feedback granularities
- Shows learning in real-time
- Reinforces that user is training, not being evaluated

Phase 5: Network Introduction (2 minutes)

Objective: Explain next phase, introduce verification, set expectations

Experience:

NETWORK READY

Congratulations! [Agent Name] has completed basic training.

CURRENT STATUS:

 Sync Level: 88% (Bronze Tier)

 Missions Completed: 1

 Total XP: 375

 Network Reputation: Building...

WHAT'S NEXT:

Daily Challenges

Quick 2-5 minute tasks to maintain and improve sync.

Earn XP and strengthen [Agent Name]'s capabilities.

Verification Missions

Help verify outputs from other agents in the network.

Earn Verification Credits and boost network quality.

Advanced Training

Unlock new capabilities as you reach Silver and Gold sync.

Agent Arena (Coming Soon)

Compete in friendly challenges. Showcase skills.

You can interact whenever you have time—no pressure.

[Agent Name] is ready when you are.


[\[View Daily Challenge\]](#) [\[Start Verification\]](#) [\[Agent Profile\]](#)

Design Notes:

- Celebrates completion
 - Shows clear progression path
 - Multiple engagement options
 - Low-pressure framing
 - Call-to-action without forcing
-

Post-Onboarding: Engagement Loops

Daily Challenge Structure


 DAILY CHALLENGE (Dec 26)

Quick sync check for [Agent Name]

Today's Scenario: [Relevant to archetype]

Time: ~3 minutes

Reward: 75 XP + Coherence boost

Status:  8 hours remaining | 47 completions today

[\[Start Challenge\]](#) [\[Skip Today\]](#)

Verification Mission Flow

✓ VERIFICATION REQUEST

Another agent completed a task. Review their output:

TASK: [Description]


AGENT OUTPUT: [The work]

YOUR EVALUATION: [Rubric]

Earn: 50 XP + 1 Verification Credit




[\[Review Output\]](#) [\[See My Stats\]](#) [\[Learn More\]](#)

Agent Profile Access


 [AGENT NAME] - THE SCOUT


Sync Level: 88%


Rank: Bronze Sync

Network Reputation:    ☆ ☆

CAPABILITIES:

 Source Evaluation: Lv 3



 Pattern Recognition: Lv 2

 Speed Analysis: Lv 1

STATS:

- Missions Completed: 1
- Predictions Made: 14
- Accuracy Rate: 79%
- Verifications: 0

ACHIEVEMENTS:

-  First Contact
-  Pattern Recognition
-  Calibrated

[\[Daily Challenge\]](#) [\[Advanced Training\]](#) [\[Leaderboard\]](#)

Technical Requirements Summary

Data Storage Needs

- User ID + Agent ID mapping
- Archetype selection
- Agent name
- Sync score (0-100%)
- Prediction history (questions, agent predictions, user responses)
- Mission completion records
- XP total
- Verification credits
- Achievement flags

Discord Bot Commands

- `/quest` or `/start` - Begin onboarding
- `/agent` - View agent profile
- `/challenge` - Today's daily challenge
- `/verify` - Start verification mission
- `/sync` - Quick sync check

AI Agent Requirements

- Prediction engine based on training data
- Archetype-specific knowledge bases
- Task execution capability (with user verification)
- Learning from feedback loop
- Confidence scoring

UI Components

- Progress bars (sync level)
 - Card displays (scenarios, sources)
 - Ranking interfaces (drag-drop)
 - Feedback forms (scaled, text)
 - Stat displays (agent profile)
-

Success Metrics

Onboarding Completion

- **Target:** 70% of users who start complete Phase 3
- **Measure:** Users reaching "Synchronization Complete"

Time to Value

- **Target:** Under 15 minutes to first mission
- **Measure:** Time from /quest to Phase 4 complete

Engagement Quality

- **Target:** 60% of users return within 48 hours
- **Measure:** Users completing second daily challenge

Sync Quality

- **Target:** Average 75%+ sync after onboarding
 - **Measure:** Prediction accuracy in first verification tasks
-

Open Questions for Development

1. **Prediction Engine:** Rule-based or ML model for agent predictions during onboarding?
 2. **Content Generation:** Static question bank or dynamic AI-generated scenarios?
 3. **Feedback Loop:** How does user correction in Phase 4+ update agent behavior?
 4. **Difficulty Scaling:** How do missions increase in complexity as sync improves?
 5. **Social Features:** When/how do users see other agents' work?
-

Ready for wireframe creation and technical specification