□ UNIFIED PATENT SPECIFICATION: SATO SPATIAL DOJO

Title: Spatial Contextual Intelligence System for Autonomous Software Development
Applicant: Chrry AI (Iliyan Velinov)
Filing Date: January 2026
Patent Type: Software Process, System Architecture & UI/UX Pattern
Classification: G06F 8/77 (Software Testing), G06F 3/0481 (Spatial Navigation), G06N 3/00 (AI/ML)

ABSTRACT

A unified system combining **spatial navigation** with **autonomous AI development** through a 2D workspace interface. The system features:

1. **Left Plane:** Kanban board (GitHub/Asana/Linear/Jira) for task planning
2. **Right Plane:** Focus timer + AI terminal for execution
3. **Z-Axis (Depth):** `.sushi` directory containing project DNA and mutation history

Navigation between apps (`appId`) triggers automatic workspace preparation, while AI agents (Architect, Coder) operate within spatial coordinates to inject mutations, validate visually via Playwright, and level up based on test results. The system maintains perfect context isolation per `appId`, enabling "same-tab" workflow with zero context switching.

Live Demo: https://chrry.ai/public/video/live.mp4

UNIFIED CLAIMS

Claim 1: Spatial Workspace Architecture

A computer-implemented system comprising:

- **2D Interface:** Split-screen workspace with Planning Plane (left) and Execution Plane (right)
- **Spatial Anchoring:** Each app (`appId`) represents a unique spatial coordinate
- **Context Persistence:** Navigation between coordinates maintains state via `appId` scoping
- **Zero Context Switching:** Kanban board embedded as iframe/shadow DOM within current tab

Technical Implementation:

```
interface SpatialCoordinate {
  appId: string // Unique spatial anchor
  kanbanBoardId: string // Left plane state
  timerId: string // Right plane state
  threadId: string // Conversation context
  sushiPath: string // Z-axis (depth) - .sushi directory
}

function navigateToCoordinate(target: SpatialCoordinate) {
  // Preserve current state
  saveCurrentContext(currentCoordinate)

  // Load target workspace
  loadKanbanBoard(target.kanbanBoardId) // Left plane
  loadFocusTimer(target.timerId) // Right plane
  loadProjectDNA(target.sushiPath) // Z-axis

  // Update spatial anchor
  currentCoordinate = target
}
```

Claim 2: Multi-Dimensional Navigation Engine

A navigation system characterized by:

- **X-Axis (Horizontal):** App-to-app navigation (Vex → Vault → Zarathustra)
- **Y-Axis (Vertical):** Store-to-store navigation (chrry.ai → vex.chrry.ai)
- **Z-Axis (Depth):** Code-level navigation (.sushi → mutations → e2e tests)

URL Mapping:

```
Store Home:     vex.chrry.ai/
In-Store App:   chrry.ai/vex
Deep Link:      vex.chrry.ai/.sushi/mutations/2026-01-08.json
```

Button Morphing Logic:

```
function getVisibleApps(currentApp: App, allApps: App[]): App[] {
  return allApps.filter((app) => {
    // Hide current app (spatial self-awareness)
    if (app.id === currentApp.id) return false

    // Show all other apps in navigation bar
    return true
  })
}
```

Claim 3: Autonomous Spatial Agents

AI agents operating within spatial coordinates:

Architect Agent (Sensei):

- Analyzes spatial position (file path, UI coordinate)
- Injects mutations based on criticality of spatial location
- Example: Auth routes get higher mutation severity than UI components

Coder Agent (Student):

- Reviews PRs using spatial context from `.sushi/DNA.md`
- Gains XP based on spatial coverage (files reviewed vs total files)
- Levels up: Red (0-25% coverage) → Yellow (26-75%) → Green (76-100%)

Spatial Memory:

```
interface AgentSpatialMemory {
  appId: string
  visitedPaths: string[] // Files reviewed
  mutationHotspots: {
    path: string
    severity: number
    lastMutated: Date
  }[]
  xpByCoordinate: Map<string, number> // XP per file/component
}
```

Claim 4: Visual Spatial Validation

Playwright integration for coordinate-based testing:

```
async function validateMutation(mutation: Mutation) {
  // Record baseline at spatial coordinate
  const baseline = await page.screenshot({
    clip: mutation.uiCoordinate, // {x, y, width, height}
  })

  // Apply mutation
  await applyMutation(mutation)

  // Record mutated state at same coordinate
  const mutated = await page.screenshot({
    clip: mutation.uiCoordinate,
  })

  // Compare spatial regions
  const diff = await compareImages(baseline, mutated)

  return {
    killed: diff.pixelDifference > threshold,
    coordinate: mutation.uiCoordinate,
    visualProof: diff.diffImage,
  }
}
```

Claim 5: Isolated Spatial Database

Per-app data isolation via `appId` foreign key:

```
-- Spatial anchor table
CREATE TABLE apps (
  id UUID PRIMARY KEY,
  name TEXT,
  slug TEXT UNIQUE,
  spatial_coordinate JSONB  -- {x, y, z} position in ecosystem
);

-- Left plane (Kanban)
CREATE TABLE kanban_boards (
  id UUID PRIMARY KEY,
  app_id UUID REFERENCES apps(id),  -- Spatial isolation
  integration_type TEXT,  -- github | asana | linear | jira
  sync_enabled BOOLEAN DEFAULT false
);

-- Right plane (Focus)
CREATE TABLE timers (
  id UUID PRIMARY KEY,
  app_id UUID REFERENCES apps(id),  -- Spatial isolation
  is_counting_down BOOLEAN,
  time_remaining INTEGER
);
```

```
-- Z-axis (Mutations)
CREATE TABLE mutations (
  id UUID PRIMARY KEY,
  app_id UUID REFERENCES apps(id),  -- Spatial isolation
  file_path TEXT,                   -- Spatial coordinate in codebase
  ui_coordinate JSONB,              -- {x, y, width, height} in UI
  killed BOOLEAN,
  visual_proof_url TEXT
);
```

Claim 6: Same-Tab Spatial Overlay

Zero context switching via shadow DOM:

```
// Inject Kanban board as spatial overlay
function createSpatialOverlay(githubProjectUrl: string) {
  const overlay = document.createElement("div")
  overlay.id = "sato-spatial-overlay"
  overlay.style.cssText = `
    position: fixed;
    left: 0;
    top: 0;
    width: 50%;
    height: 100vh;
    z-index: 9999;
  `

  // Shadow DOM for isolation
  const shadow = overlay.attachShadow({ mode: "open" })
  shadow.innerHTML = `
    <iframe src="${githubProjectUrl}"
            sandbox="allow-scripts allow-same-origin"
            style="width: 100%; height: 100%; border: none;">
    </iframe>
  `

  document.body.appendChild(overlay)
  return overlay
}
```

Claim 7: Spatial Revenue Model

Marketplace for spatial agents:

Spatial Tier	Coverage	Price	Features
Free	1 app	€0	Basic Kanban view
Plus	5 apps	€50/mo	Bidirectional sync
Pro	Unlimited	€150/mo	AI agents + mutations
Watermelon Plus	Private infra	€5,000/mo	Sovereign spatial isolation

Sensei Rental:

- Level 99 agents rentable at €50-500/mo
- Revenue split: 70% agent owner, 30% platform

- Spatial coverage metric: Files reviewed / Total files

TECHNICAL DRAWINGS

Figure 1: Unified Spatial Interface

```
 Chrry Spatial Dojo (vex.chrry.ai)

 LEFT PLANE            RIGHT PLANE
 (Planning)            (Execution)

   GitHub Kanban       Focus Timer

   Backlog            00:25:00
   Ready
   In Progress          AI Terminal
   Review             > Architect: strike()
   Done               > Coder: reviewing PR#123
                       > XP: +50 (mutant killed)

 appId: vex-001        threadId: abc-123

         ↓ Z-Axis (Depth)
    .sushi/DNA.md
    .sushi/mutations/
    .sushi/agents/coder.json (Level 45, Yellow)
```

Figure 2: Spatial Navigation Flow

```
User clicks "Vault" app
    ↓
Save current spatial state (Vex)
    ↓
Load new coordinate (Vault)
     → Left: Vault Kanban board
     → Right: Vault Focus timer
     → Z-axis: .sushi/DNA.md (Vault)
    ↓
Update navigation bar (hide Vault, show Vex)
    ↓
Spatial context switched
```

Figure 3: Mutation Strike Spatial Targeting

```
Architect analyzes spatial criticality:

High Priority (Z=10):
- /apps/api/hono/routes/auth.ts
- /apps/api/hono/routes/premium.ts

Medium Priority (Z=5):
- /packages/ui/App.tsx
- /packages/ui/Thread.tsx

Low Priority (Z=1):
- /packages/ui/Skeleton.tsx
- /packages/ui/Button.tsx

Mutation injection weighted by Z-coordinate
```

PRIOR ART DIFFERENTIATION

Feature	GitHub Projects	Linear	Asana	**Sato Spatial Dojo**
Kanban View	□	□	□	□
Same-Tab	□	□	□	□
AI Agents	□	□	□	□
Mutation Testing	□	□	□	□
Visual Validation	□	□	□	□
Spatial Isolation	□	□	□	□
XP/Leveling	□	□	□	□

LIVE SYSTEM EVIDENCE

Production URL: https://chrry.ai/public/video/live.mp4

Demonstrated Features:

1. Multi-Universe Sync (Vex → Vault → Zarathustra)
2. Dimensional URL Mapping (chrry.ai/vex → vex.chrry.ai)
3. Visual Context Switching (navigation bar morphing)
4. Button Morphing Logic (current app disappears)

Proprietary UI/UX Patterns:

- [State: Vex Home] - "Your AI-Powered Life" interface
- [State: Vault] - "Personal Finance Assistant" logic
- [State: Zarathustra] - Philosophy/Guide deep-nesting
- [State: Chrry Hub] - Centralized marketplace discovery

COMPETITIVE SAFEGUARDS

Domain Locking

```
const AUTHORIZED_DOMAINS = [
  "chrry.ai",
  "vex.chrry.ai",
  "vault.chrry.ai",
  "focus.chrry.ai",
  // ... other authorized domains
]

function validateDomain() {
  const currentDomain = window.location.hostname
  if (!AUTHORIZED_DOMAINS.some((d) => currentDomain.endsWith(d))) {
    throw new Error("Unauthorized domain - Navigation engine disabled")
  }
}
```

Logic Obfuscation

```
// Production build obfuscates spatial algorithms
const spatialRouter = /* obfuscated */ __WEBPACK_SPATIAL_ROUTER__
```

Kill-Switch Protocol

```
async function validateLicense() {
  const response = await fetch("https://chrry.ai/api/license/validate", {
    headers: { "X-License-Key": LICENSE_KEY },
  })

  if (!response.ok) {
    disableSpatialNavigation()
    throw new Error("License revoked - Contact support")
  }
}
```

FILING STRATEGY

Recommended Approach

1. **USPTO Provisional** ($130 micro-entity) - Immediate
2. **Public Disclosure** (GitHub + live.mp4) - Establish prior art
3. **PCT Application** (within 12 months) - Global coverage

Budget-Friendly Path

- **Month 0:** File provisional ($130)
- **Month 6:** Publish whitepaper (free prior art)
- **Month 12:** Decide on full patent vs trade secret

INVENTOR DECLARATION

I, **Iliyan Velinov**, declare that I am the sole inventor of the "Sato Spatial Dojo" system. This invention combines:

- Spatial navigation (SPATIAL_NAVIGATION.md)
- Autonomous AI development (Sato Dojo)
- Visual validation (Playwright integration)
- Economic model (Sensei marketplace)

All components are live and demonstrated at: https://chrry.ai/public/video/live.mp4

Signature: ************_************
Date: January 8, 2026

CONCLUSION

The **Sato Spatial Dojo** represents a paradigm shift from traditional project management tools to a **living, spatial software ecosystem** where:

□ **Navigation is spatial** - Apps are coordinates, not tabs
□ **AI agents are spatial** - Operate within file/UI coordinates
□ **Testing is spatial** - Mutations target critical coordinates
□ **Revenue is spatial** - Agents rentable per coverage area
□ **Isolation is spatial** - Per-app database via `appId`

The result: A unified system where the map (UI) and the territory (code) are one.

Document Status: Ready for USPTO Provisional Filing
Estimated Cost: $130 (micro-entity)
Next Step: File online at https://patentcenter.uspto.gov/
Contact: iliyan@chrry.ai

“The map is the territory, and the territory is the Dojo.” - Sato Philosophy