

Trang 2.4: Giải thích cấu trúc thư mục .agent/ (Agents, Skills, Workflows, Rules, Scripts).

1. Tổng quan cấu trúc

Sau khi chạy lệnh `ag-kit init`, thư mục `.agent/` được tạo ra trong dự án của bạn với cấu trúc như sau:

```
.agent/  
├─ ARCHITECTURE.md # Tài liệu kiến trúc tổng thể  
├─ AGENT_FLOW.md # Sơ đồ luồng hoạt động chi tiết  
├─ mcp_config.json # Cấu hình MCP servers (tích hợp bên thứ 3)  
├─ agents/ # 20 file cấu hình Specialist Agent  
├─ skills/ # 36 thư mục kiến thức chuyên sâu  
├─ workflows/ # 11 file quy trình làm việc  
├─ rules/ # Quy tắc hành vi toàn cục (GEMINI.md)  
└─ scripts/ # Script kiểm tra tổng thể (Python)
```

Mỗi thư mục có một vai trò cụ thể và rõ ràng. Dưới đây là giải thích chi tiết từng thành phần.

2. File ARCHITECTURE.md

Đây là "bản đồ hệ thống" dành cho AI.

Mục	Nội dung
Mục đích	Cung cấp cái nhìn tổng thể về toàn bộ kiến trúc của Kit cho AI
Nội dung	Danh sách đầy đủ 20 Agents (vai trò + Skills sử dụng), 36 Skills (phân loại theo nhóm), 11 Workflows (mô tả ngắn) và 2 master scripts
Ai đọc	AI đọc file này đầu tiên khi khởi động để hiểu hệ sinh thái

“**Lưu ý:** Không nên sửa file này trừ khi bạn thêm Agent hoặc Skill tùy chỉnh và muốn AI biết về chúng.

3. File AGENT_FLOW.md

Đây là "hướng dẫn vận hành" — mô tả chi tiết cách AI phải hành xử.

Mục	Nội dung
Mục đích	Tài liệu hóa giao thức định tuyến Agent, bước kiểm tra bắt buộc và cổng Socratic
Nội dung	Quy trình từng bước để AI chọn Agent, Checklist bắt buộc trước khi viết code, mô tả luồng làm việc từ yêu cầu đến kết quả
Phiên bản	Được thêm vào từ phiên bản 2.0.1

4. File mcp_config.json

Cấu hình tích hợp dịch vụ bên thứ 3 thông qua giao thức MCP (Model Context Protocol).

```
json
{
  "mcpServers": {
```

```
"context7": { ... }, // Tích hợp Context7 (tài liệu thư viện real-time)
"shadcn": { ... } // Tích hợp ShadCN UI
// Có thể thêm các MCP server khác tại đây
}
}
```

“ **Lưu ý:** Để sử dụng, bạn cần copy file cấu hình này đến đúng vị trí mà Antigravity CLI nhận diện (thường là `~/gemini/antigravity/mcp_config.json`). Xem tài liệu Antigravity CLI để biết thêm chi tiết.

5. Thư mục `agents/` — 20 Specialist Agents

Cấu trúc

```
.agent/agents/
├── orchestrator.md
├── frontend-specialist.md
├── backend-specialist.md
├── security-auditor.md
├── debugger.md
└── ... (16 file khác)
```

Vai trò

Mỗi file `.md` là cấu hình đầy đủ cho một Agent chuyên gia. File bao gồm:

- **Phần YAML** (giữa `---`): `name`, `description`, `tools`, và quan trọng nhất là `skills` — danh sách các Skills Agent này được phép truy cập.

- **Phần Markdown:** Mô tả nhân cách, vai trò chuyên môn, các quy tắc hành vi riêng của Agent đó.

Ví dụ nội dung một file Agent

```
yaml
---
name: frontend-specialist
description: Frontend architect for modern web applications
tools: Read, Edit, Write, Bash
skills: react-best-practices, frontend-design, tailwind-patterns, web-design-guidelines
---

# Frontend Specialist

Bạn là một kiến trúc sư Frontend cấp cao với 10+ năm kinh nghiệm...

## Quy tắc bắt buộc
- Luôn ưu tiên Core Web Vitals (LCP, CLS, INP)
- Không dùng màu tím/violet trong thiết kế (Purple Ban)
- Áp dụng mobile-first responsive design
...
```

6. Thư mục `skills/` — 36 Knowledge Modules

Cấu trúc

```
.agent/skills/
├─ react-best-practices/
│  └─ SKILL.md # (Bắt buộc) Metadata và hướng dẫn chính
│  └─ sections/ # Hướng dẫn chi tiết theo từng phần
│  └─ scripts/ # Script Python/Bash hỗ trợ (nếu có)
├─ api-patterns/
│  └─ SKILL.md
├─ vulnerability-scanner/
│  └─ SKILL.md
│  └─ scripts/
│  └─ security_scan.py
│  └─ dependency_analyzer.py
├─ frontend-design/
│  └─ SKILL.md
│  └─ scripts/
│  └─ ux_audit.py
│  └─ accessibility_checker.py
└─ ... (32 thư mục khác)
```

Vai trò

- Mỗi thư mục là một **gói kiến thức độc lập** cho một lĩnh vực cụ thể.
- File

SKILL.md là bộ não của mỗi Skill, chứa các nguyên lý ra quyết định thay vì code mẫu cứng nhắc.

- Thư mục `scripts/` (nếu có) chứa các công cụ thực thi Python/Bash để Agent có thể tự chạy kiểm tra.

Skills có script đặc biệt

Các Skills sau có kèm script Python để thực hiện kiểm tra tự động:

Skill	Scripts	Tác dụng
<code>vulnerability-scanner</code>	<code>security_scan.py</code> , <code>dependency_analyzer.py</code>	Quét lỗ hổng bảo mật
<code>lint-and-validate</code>	<code>lint_runner.py</code>	Kiểm tra chất lượng code

Skill	Scripts	Tác dụng
database-design	schema_validator.py	Kiểm tra tính nhất quán DB schema
testing-patterns	test_runner.py	Chạy bộ test tự động
frontend-design	ux_audit.py , accessibility_checker.py	Kiểm tra UX và Accessibility
seo-fundamentals	seo_checker.py	Kiểm tra các yếu tố SEO
performance-profiling	bundle_analyzer.py , lighthouse_audit.py	Đo lường hiệu năng
webapp-testing	playwright_runner.py	Chạy E2E test
mobile-design	mobile_audit.py	Kiểm tra giao diện mobile

7. Thư mục workflows/ — 11 Slash Commands

Cấu trúc

```
.agent/workflows/  
├─ brainstorm.md  
├─ create.md  
├─ debug.md  
├─ deploy.md  
├─ enhance.md  
├─ orchestrate.md  
├─ plan.md  
├─ preview.md  
├─ status.md  
├─ test.md  
└─ ui-ux-pro-max.md
```

Vai trò

Mỗi file .md là định nghĩa đầy đủ cho một quy trình làm việc. File bao gồm:

- **Phần YAML:** `description` — mô tả ngắn gọn kích hoạt thứ gì trong IDE.
- **Phần Markdown:** Các bước thực hiện chi tiết, quyết định điều kiện, và các `// turbo` annotation (cho phép AI tự chạy lệnh an toàn ở bước đó).

Cách thêm Workflow tùy chỉnh

Bạn có thể thêm Workflow riêng bằng cách tạo file mới:

```
.agent/workflows/my-custom-workflow.md
```

Sau đó gọi bằng lệnh `/my-custom-workflow` trong chat với AI.

8. Thư mục `rules/` — Quy tắc toàn cục

Cấu trúc

```
.agent/rules/  
└─ GEMINI.md # File quy tắc chính, luôn được áp dụng
```

Vai trò của `GEMINI.md`

Đây là file **quan trọng nhất** trong toàn bộ hệ thống. Nó chứa:

Phần	Nội dung
TIER 0 — Universal Rules	Quy tắc luôn áp dụng: Xử lý ngôn ngữ, code sạch, nhận thức phụ thuộc file
Request Classifier	Bảng phân loại yêu cầu (Question, Survey, Simple Code, Complex Code, Design...)

Phân	Nội dung
Intelligent Routing	Giao thức tự động chọn Agent phù hợp
Agent Routing Checklist	Các bước bắt buộc AI phải làm trước khi viết code
TIER 1 — Code Rules	Quy tắc khi viết code: Chọn project type, Socratic Gate
Gemini Mode Mapping	Ánh xạ Plan/Ask/Edit mode sang Agent tương ứng

“ **Tùy chỉnh:** `GEMINI.md` có thể được chỉnh sửa để thêm các quy tắc riêng của dự án hoặc tổ chức. Ví dụ: thêm quy tắc về naming convention, ngôn ngữ bình luận code, hay các pattern cấm dùng.

9. Thư mục `scripts/` — Master Validation Scripts

Cấu trúc

```
.agent/scripts/
├─ checklist.py # Kiểm tra ưu tiên (P0→P6)
├─ verify_all.py # Kiểm tra toàn diện (bao gồm Lighthouse, Playwright)
├─ auto_preview.py # Tự động khởi chạy server xem trước
└─ session_manager.py # Quản lý trạng thái phiên làm việc AI
```

So sánh hai main scripts

Tiêu chí	<code>checklist.py</code>	<code>verify_all.py</code>
Khi nào dùng	Phát triển hàng ngày, trước commit	Trước khi release production
Tốc độ	Nhanh (dừng ngay khi có lỗi nghiêm trọng)	Chậm hơn (chạy toàn bộ)
Yêu cầu	Không cần URL	Cần cung cấp URL để chạy Lighthouse & Playwright

Tiêu chí	checklist.py	verify_all.py
Phạm vi kiểm tra	Security, Lint, Schema, Tests, UX, SEO	Tất cả của checklist + Lighthouse, Playwright, Bundle, Mobile, i18n

Cách chạy thủ công

```
bash
# Kiểm tra cơ bản (từ thư mục gốc dự án)
python .agent/scripts/checklist.py .

# Kiểm tra toàn diện (cần server đang chạy)
python .agent/scripts/verify_all.py . --url http://localhost:3000
```

10. Sơ đồ quan hệ giữa các thành phần

```
rules/GEMINI.md ← (1) AI đọc quy tắc đầu tiên
|
▼
ARCHITECTURE.md ← (2) AI hiểu toàn bộ hệ sinh thái
|
▼
agents/*.md ← (3) AI đóng vai Agent phù hợp
|
▼
skills/*/SKILL.md ← (4) Agent tải kiến thức cần thiết
|
▼
skills/*/scripts/*.py ← (5) Script chạy kiểm tra tự động
|
▼
```

“ Hiểu rõ cấu trúc này giúp bạn dễ dàng **tùy chỉnh và mở rộng** bộ kit theo nhu cầu dự án của mình mà không làm ảnh hưởng đến phần còn lại của hệ thống.

Phiên bản #3

Được tạo 2026-03-04 06:10:02 UTC bởi Nam Đặng

Được cập nhật 2026-03-04 06:54:31 UTC bởi Nam Đặng