



SUGCON
EUROPE 2026

Vibesitecoring: From Figma to Sitecore with Zero Lines of Code

The future is here?



Volodymyr Nikitin

CEO of EXDST

- 28 years in Software development industry
- 60+ Sitecore solutions delivered

vn@exdst.com

[@vladimirkinitin](https://twitter.com/vladimirkinitin)



Anton Tishchenko

CTO of EXDST

- Sitecore MVP (2026-2019)
- 15+ years of Sitecore experience

at@exdst.com

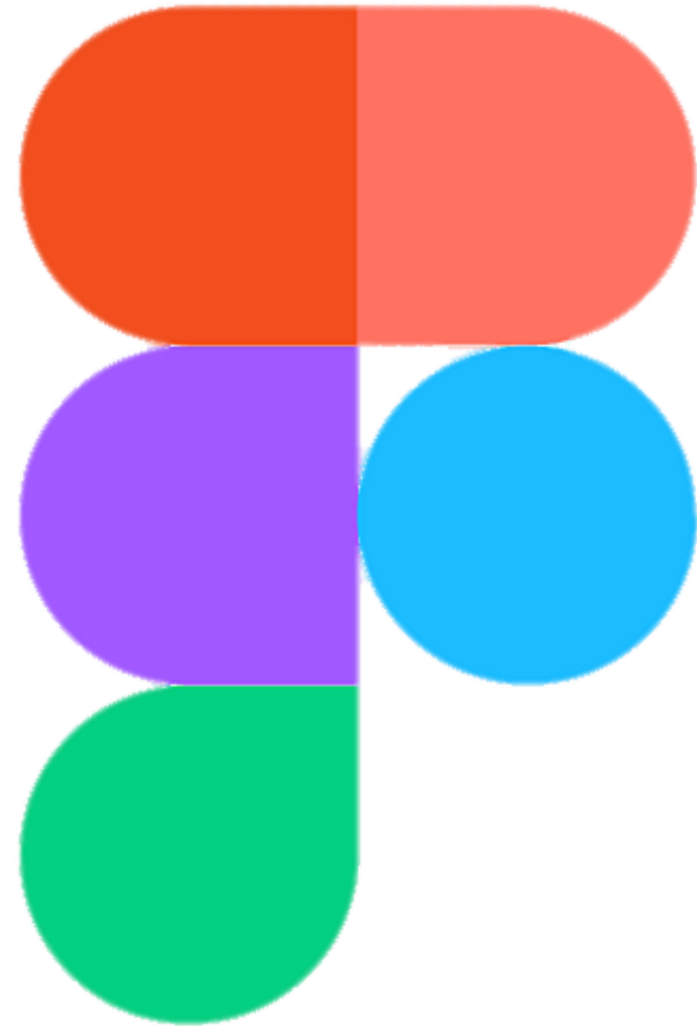
[@anton-tishchenko](https://twitter.com/anton-tishchenko)

Creators of Sitecore MCP Server

Creators of Sitecore Astro JSS

<https://exdst.com>

- Figma is de facto industry standard for web design
- We do less Sitecore customizations for SitecoreAI than we did for Sitecore XM/XP
- 70% of development work on new Sitecore projects is the creation of components
- Even small optimization of the main part of development process will make big impact on the project





The Hare

10x Software developer

- Sitecore 5, 6, 7, 8, 9, 10 certified
- SitecoreAI certified
- HTML expert & CSS guru
- JavaScript pro & C# master
- 500 words per minute keyboard typing speed
- Wants fast results from AI



The Tortoise

AI Enthusiast

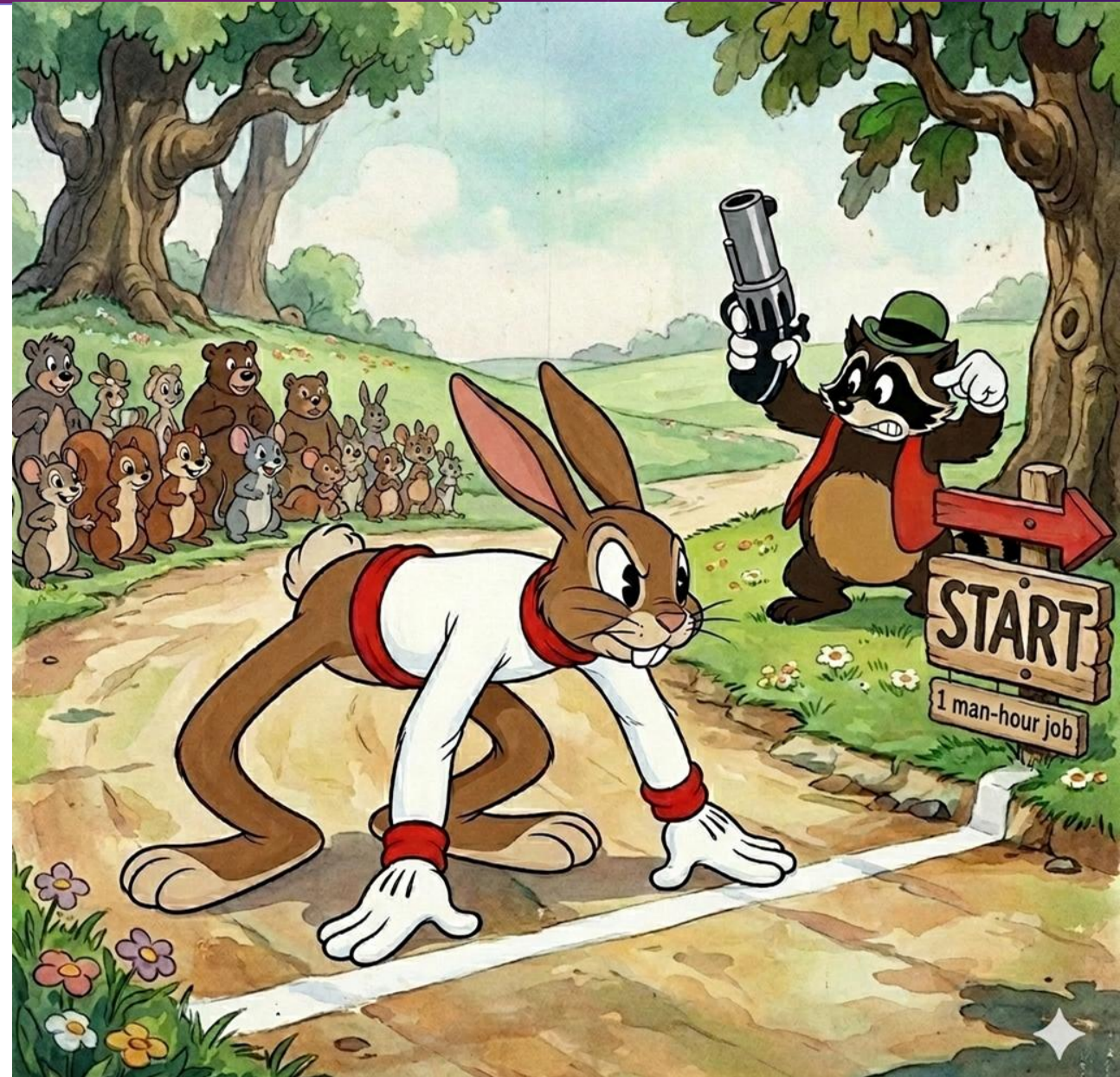
- Smart
- Stopped to write a code by himself 6 months ago
- Has 10 AI subscriptions to different services
- Believes in LLMs supremacy over the people
- Spends 200\$+ on different AI subscriptions a month

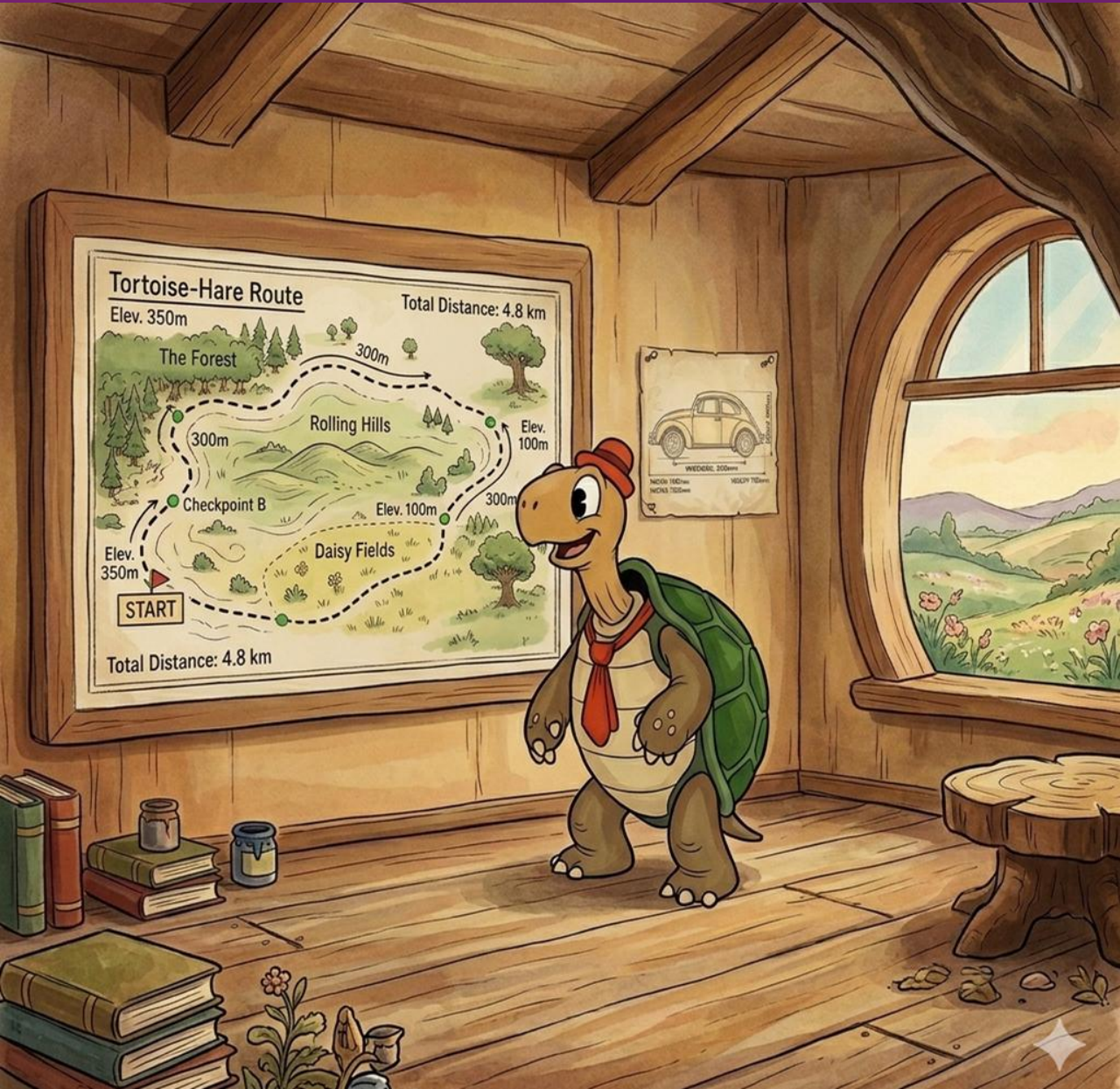


- Implementation of design could be decomposed into smaller chunks of work
- Smallest task with business value is one Sitecore rendering
- Small improvement on one small task will give big improvement on many small tasks

Implementation of React component

- Component is easy to implement
- Implementation requires skills:
 - HTML
 - CSS
 - TypeScript/JavaScript
 - React
- Figma is much more developer friendly than PSD design files





Everything is easier with Figma Dev Mode:

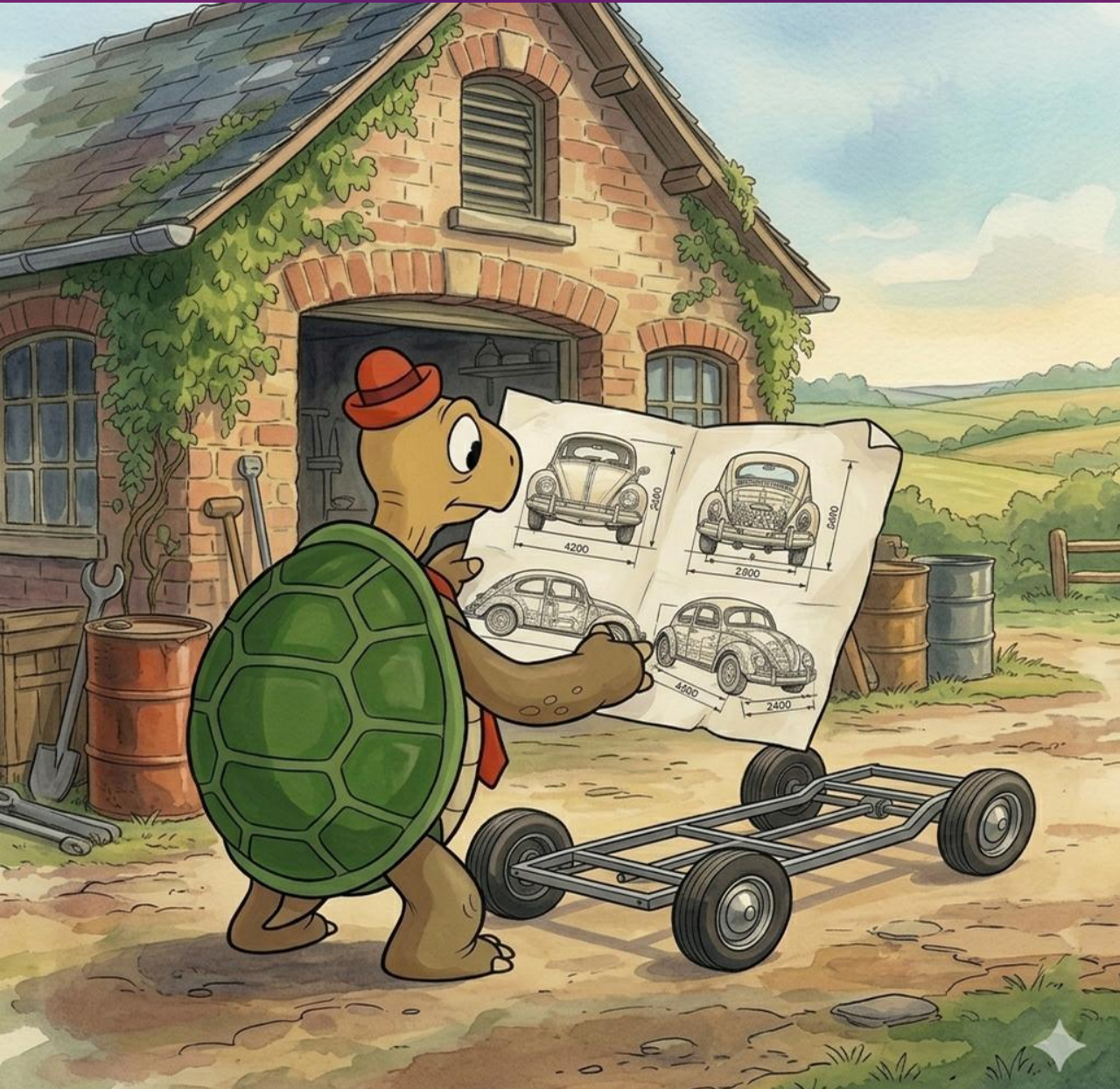
- CSS
- Auto-generated code
- Strict measurements
- Dev Mode plugins
- Better communication
- VS Code extension

AI Agents can read screenshots

You can get “fast” advantages from AI

- You can use AI only as chat
- You can share screenshots with AI
- You can write complex prompts





Figma MCP is available starting from Figma Dev seat subscription

- generate_figma_design
- **get_design_context**
- **get_variable_defs**
- get_code_connect_map
- add_code_connect_map
- **get_screenshot**
- create_design_system_rules
- get_metadata
- get_figjam

There will be bugs with AI

AI is not a magic wand that will do everything in one iteration

- Not an ideal result
- Could overcomplicate
- Set correct expectations
- Improves over time





Use it to provide feedback to your AI agent.
Installed as NPM package

- **click**, **hover**, `press_key`
- `drag`, `fill`, **fill_form**, `handle_dialog`
- `upload_file`
- `close_page`, `list_pages`, **navigate_page**,
`new_page`, `select_page`, `resize_page`
- `wait_for`, `emulate`, **evaluate_script**
- **get_console_message**,
`list_console_messages`
- `take_screenshot`, `take_snapshot` tools
- `performance_analyze_insight`,
- `get_network_request`, etc.

After frontend work is done, many things should be done in SitecoreAI (or Sitecore XM/XP)

- Datasource template, fields
- Rendering parameters template, fields
- Rendering item
- Insert options
- Test pages
- Test datasources





Developed by EXDST, installed as NPM package

- Open-source, available on GitHub <https://github.com/antonytm/mcp-sitecore-server>
- 100+ tools for all Sitecore scenarios that you can imagine
- Does not require any changes in Sitecore backend
- Based on Sitecore Item Service API, Sitecore Powershell Remoting and Sitecore GraphQL API
- Works with SitecoreAI
- Works with Sitecore XM/XP

AI Agent can do things wrong

- Do not reuse existing atomic components
- Use wrong technology for styling (e.g. SASS instead of Tailwind)
- Place code to wrong folders
- Use wrong naming convention
- Datasource template uses wrong field types
- Rendering parameters are absent
- Rendering item wrong location
- Insert options are not added
- Test pages without proper data



Give you AI enough data to make right decisions



- Use **CLAUDE.md**
- Describe your project's structure, naming conventions
- Write rules for code reuse
- Save important *IDs* and *paths*
- Deny unwanted actions
- Add chains of actions (e.g. *dotnet sitecore cli pull* on content change)
- **Improve your rules incrementally**

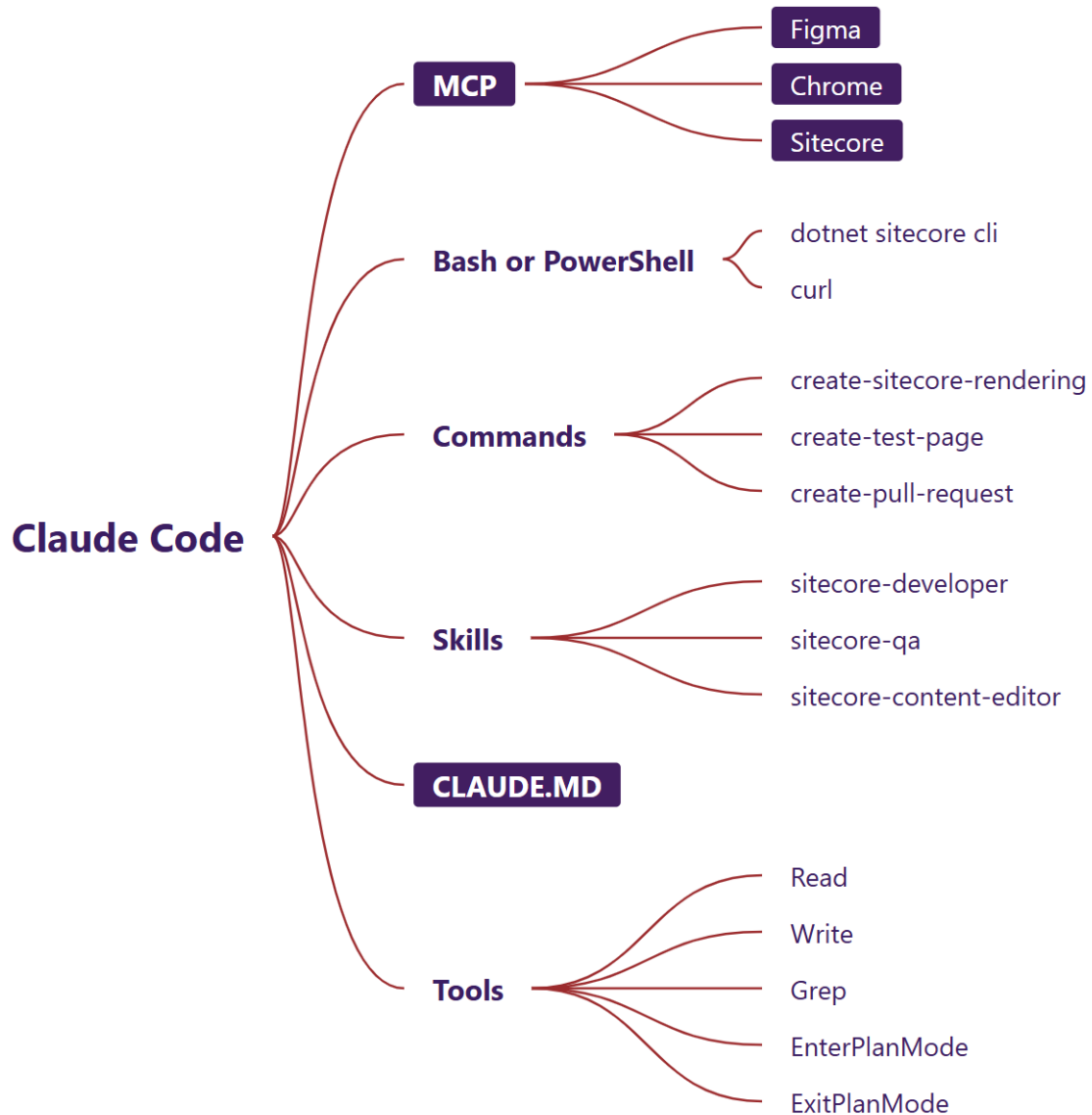
Use features available with your AI agent



For Claude Code:

- Skills
- Commands
- Hooks
- Agents
- Plan mode
- Subagents
- Memory
- Plugins
- Permissions

How it works: Claude.md + MCP Servers



```
PS C:\source\Sitecore.Demo.XMCloud.Verticals> claude
Claude Code v2.1.101
Opus 4.6 (1M context) · Claude Max
C:\source\Sitecore.Demo.XMCloud.Verticals

/mcp

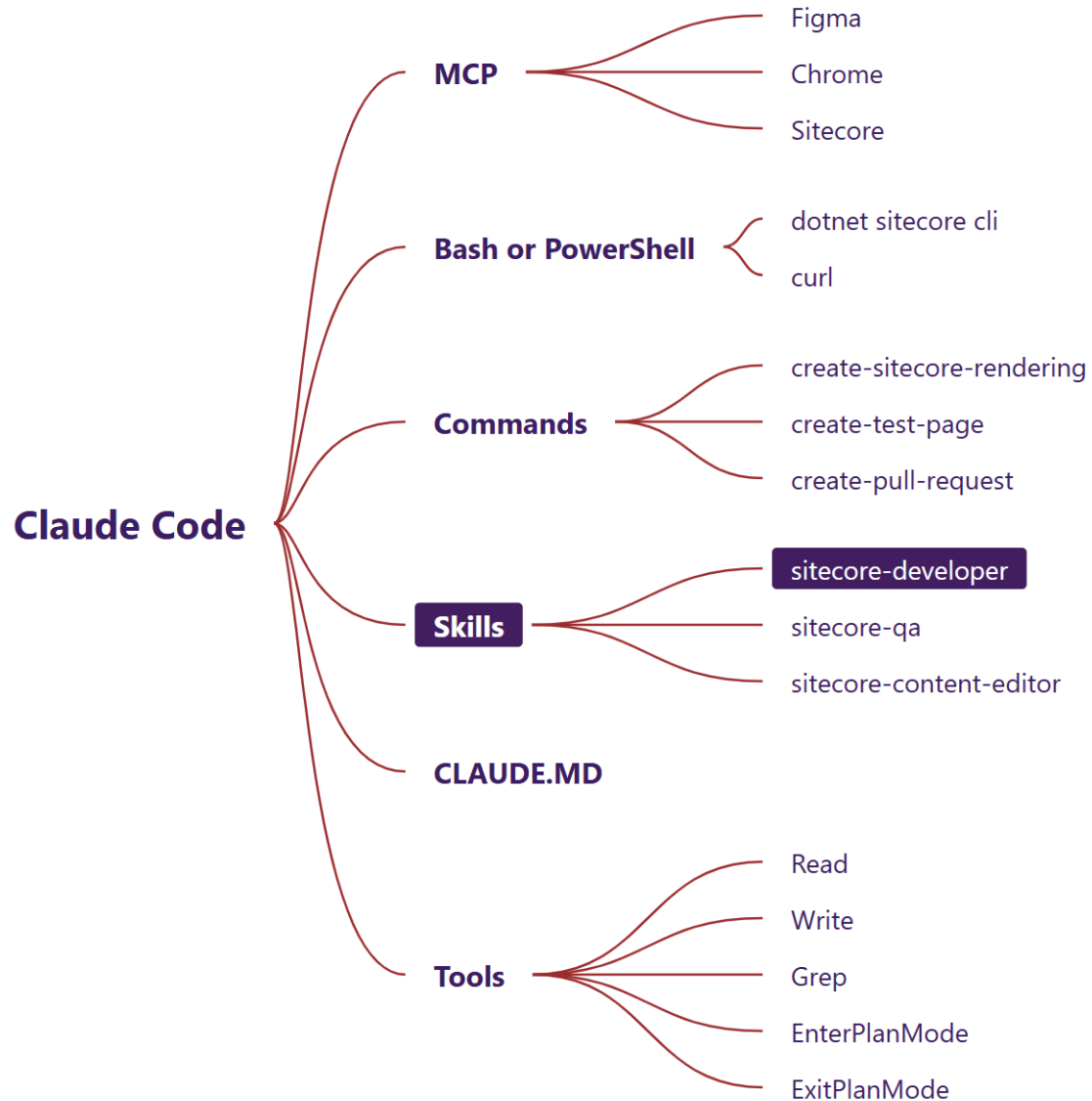
Manage MCP servers
4 servers

Project MCPs (C:\source\Sitecore.Demo.XMCloud.Verticals\.mcp.json)
> chrome-dev-tools · ✓connected
figma · ✓connected
sitecore · ✓connected

Built-in MCPs (always available)
plugin:chrome-devtools-mcp:chrome-devtools · ✓connected

https://code.claude.com/docs/en/mcp for help
↑↓ to navigate · Enter to confirm · Esc to cancel
```

How it works: Sitecore Developer Skill



```
PS C:\source\Sitecore.Demo.XMCloud.Verticals> claude
Claude Code v2.1.101
Opus 4.6 (1M context) · Claude Max
C:\source\Sitecore.Demo.XMCloud.Verticals

> /mcp
  ↳ MCP dialog dismissed

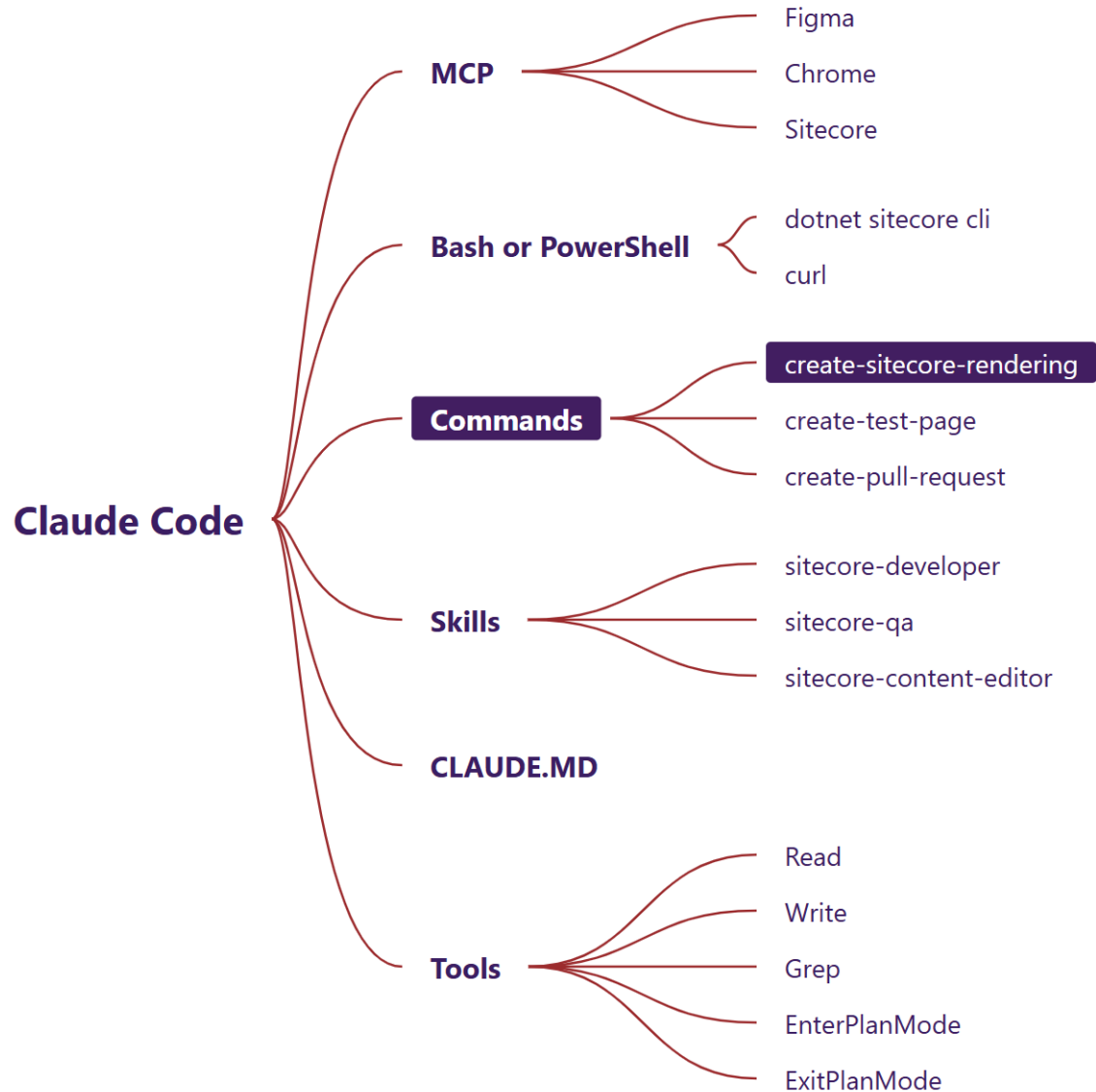
> /sitecore-developer

• I'll follow the SitecoreAI development guidelines for all subsequent Sitecore work in this session.

> |

? for shortcuts                                • medium · /effort
```

How it works: Create Sitecore Rendering Command



```
PS C:\source\Sitecore.Demo.XMCloud.Verticals> claude
Claude Code v2.1.101
Opus 4.6 (1M context) · Claude Max
C:\source\Sitecore.Demo.XMCloud.Verticals

> /clear
L (no content)

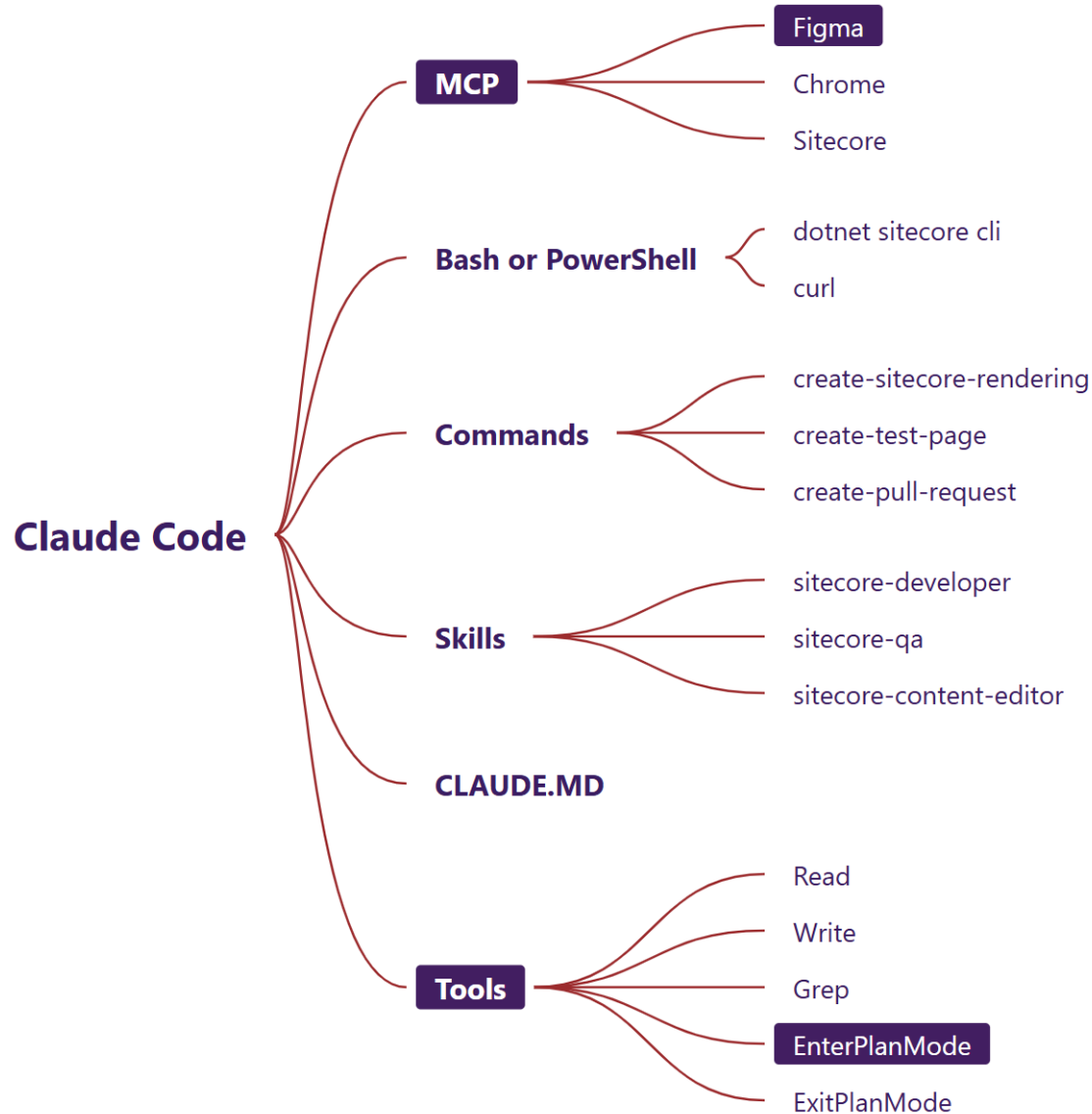
> /sitecore-developer

• I'm ready to help with SitecoreAI development work. The sitecore-developer
skill guidelines are loaded. What would you like me to do?

> /create-sitecore-rendering Implement this design from Figma.
@https://www.figma.com/design/it0RBK0KwvReot9BYbbXs/Imported-Design?node
-id=47-677&m=dev

ctrl+g to edit in Notepad
```

How it works: Read from Figma



```
Opus 4.6 (1M context) · Claude Max
C:\source\Sitecore.Demo.XMCloud.Verticals

> /sitecore-developer

• I'm ready to help with SitecoreAI development work. The skill guidelines
  are loaded.

  What would you like me to do?

> /create-sitecore-rendering Implement this design from Figma.
@https://www.figma.com/design/it0RBK0KwvReot9BYbbXs/Imported-Design?node-id=50-1100&m=dev

• Entered plan mode
  Claude is now exploring and designing an implementation approach.

• figma - get_screenshot (MCP)(nodeId: "50-1100")

• figma - get_design_context (MCP)(nodeId: "50-1100", clientFrameworks:
  "react,nextjs", clientLanguages:
  "typescript,scss", artifactType: "COMPONENT
  _WITHIN_A_WEB_PAGE_OR_APP_SCREEN",
  taskType: "CREATE_ARTIFACT")

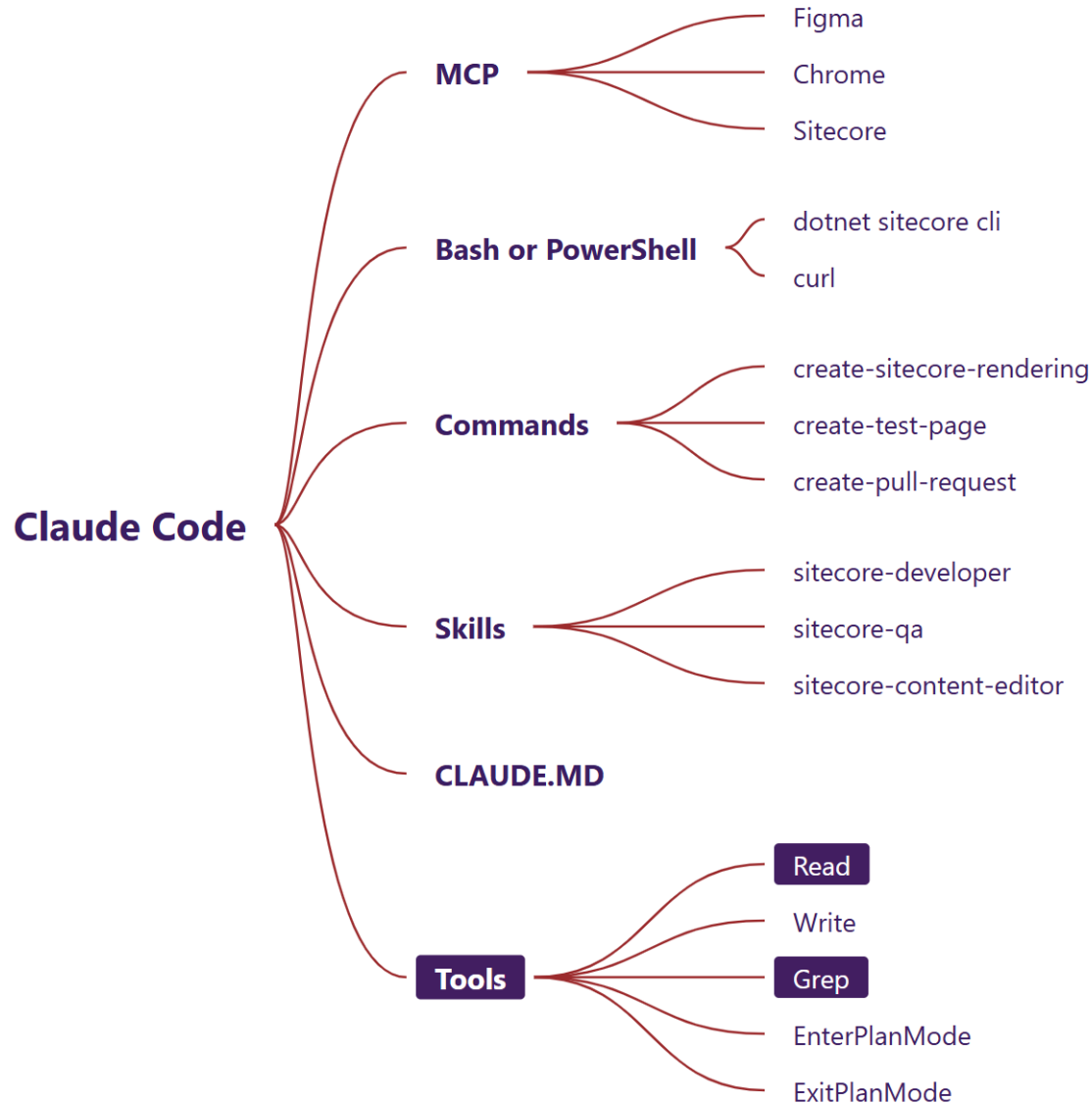
• figma - get_variable_defs (MCP)(nodeId: "50-1100", clientFrameworks:
  "react,nextjs", clientLanguages:
  "typescript,scss")

+ Zesting... (19s · ↑ 134 tokens)
  ↳ Tip: Run tasks in the cloud while you keep coding locally · clau.de/web

> |

|| plan mode on (shift+tab to cycle) · esc to interrupt
```

How it works: Review the sources



```
Claude Code
```

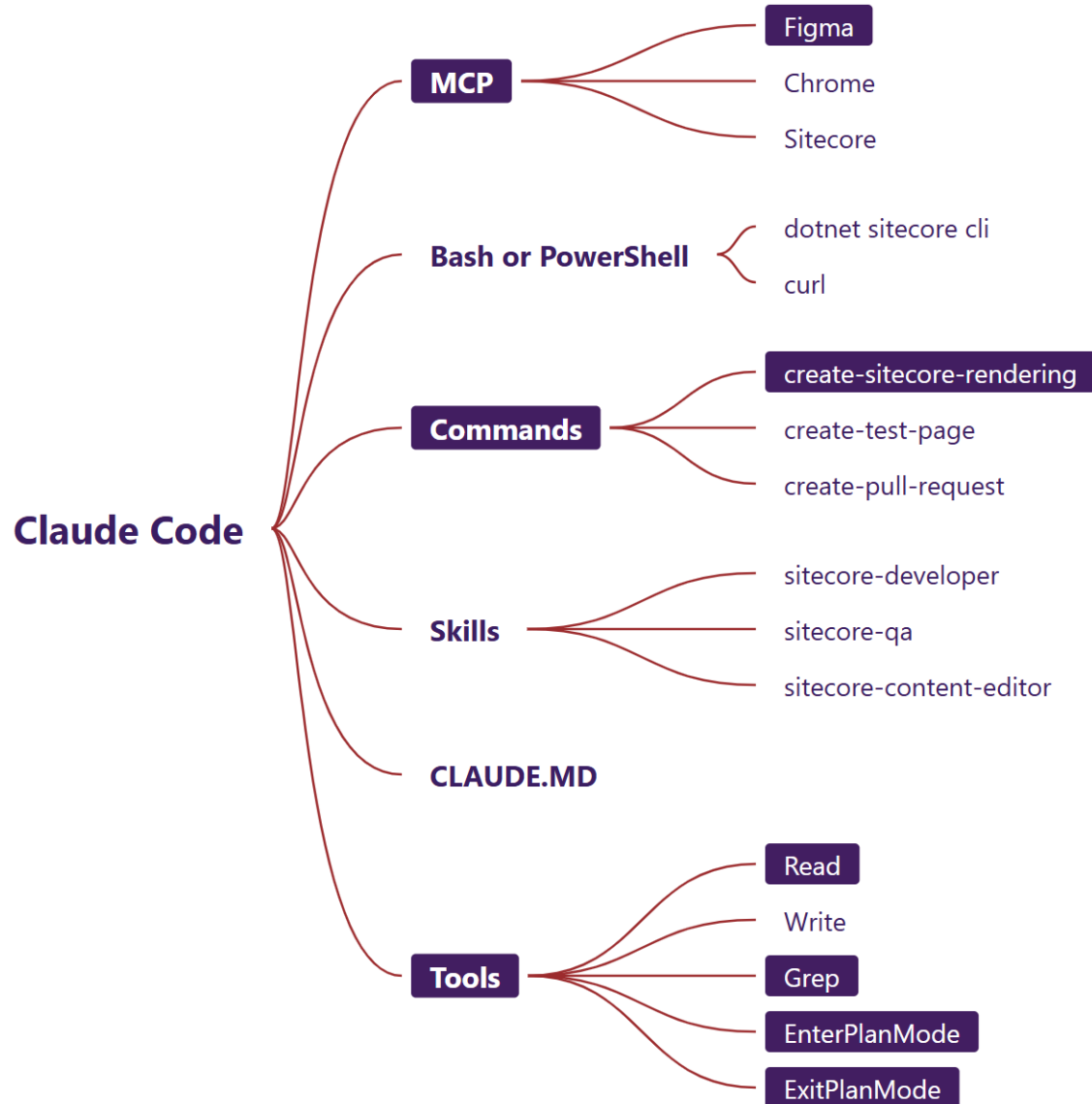
```
> /create-sitecore-rendering Implement this design from Figma.  
@https://www.figma.com/design/it0RBK0KwvReot9BYybXs/Imported-Design?node-id=50-1100&m=dev
```

- Entered plan mode
Claude is now exploring and designing an implementation approach.
- `figma - get_screenshot (MCP)(nodeId: "50-1100")`
- `figma - get_design_context (MCP)(nodeId: "50-1100", clientFrameworks: "react,nextjs", clientLanguages: "typescript,scss", artifactType: "COMPONENT_WITHIN_A_WEB_PAGE_OR_APP_SCREEN", taskType: "CREATE_ARTIFACT")`
- `figma - get_variable_defs (MCP)(nodeId: "50-1100", clientFrameworks: "react,nextjs", clientLanguages: "typescript,scss")`
- Now let me explore the codebase to understand existing component patterns.
- Explore(Explore Sitecore rendering patterns)
 - └ Bash(`find /c/source/Sitecore.Demo.XMCloud.Verticals -type d -name components | head -20`)
Running...
Search(pattern: "src/sxstarter-nextjs/src/components/**/*.*tsx")
(ctrl+b to run in background)
- Zesting... (46s · ↓ 724 tokens · thought for 2s)
 - └ Tip: Run tasks in the cloud while you keep coding locally · clau.de/web

```
> |
```

plan mode on (shift+tab to cycle) · esc to interrupt

How it works: Implementation Plan



```
Claude Code
- Open with Chrome DevTools MCP (navigate_page, take_screenshot)
- Compare against mcp__figma__get_screenshot for node 50:1100
- Iterate on SCSS / layout until visually matching (colors, spacing, pill button, left image / right text layout, responsive stack)

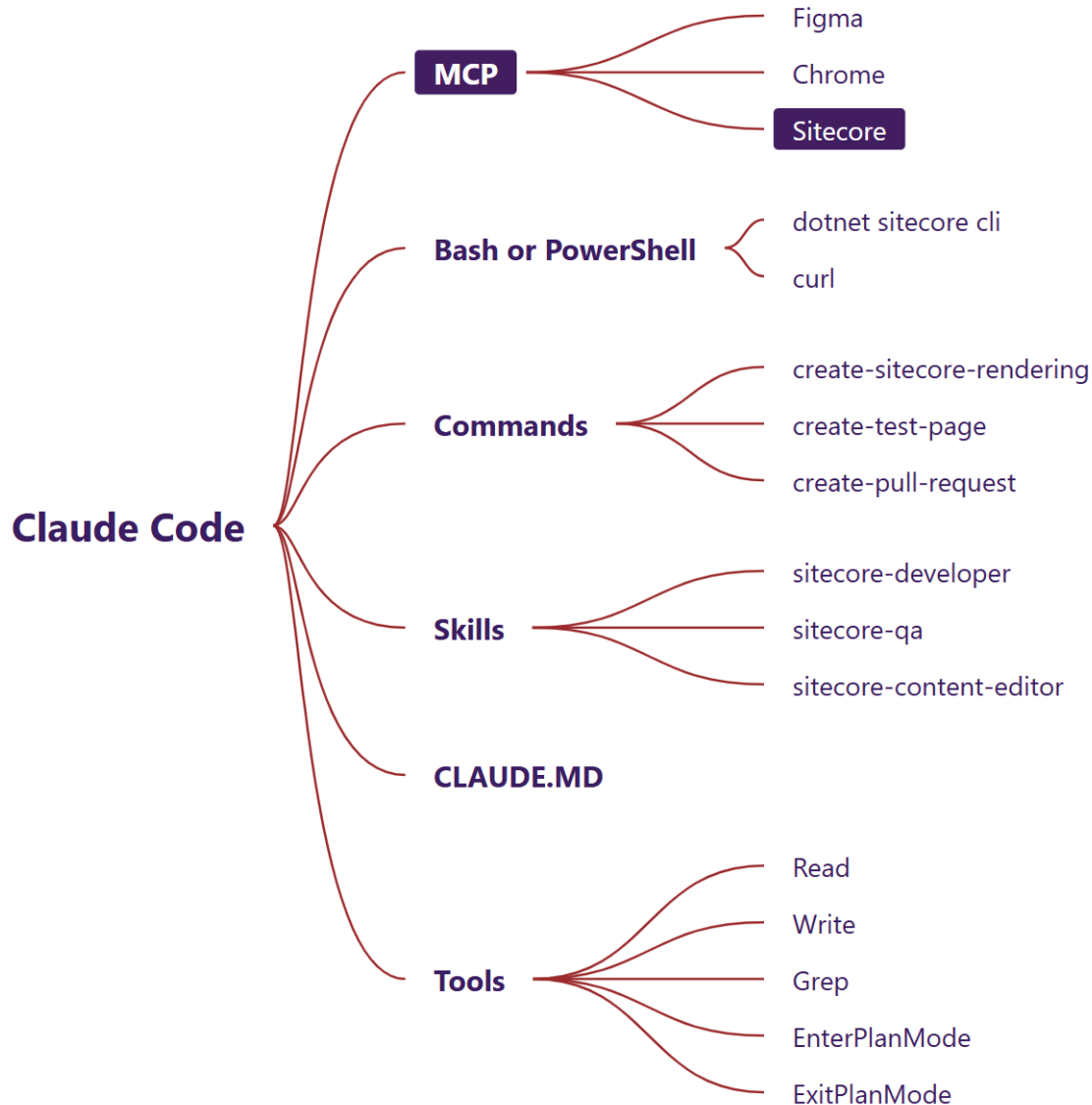
Critical files to modify / create
- Create - src/sxstarter-nextjs/src/components/PageContent/FeaturedCard.tsx
- Create - src/sxstarter-nextjs/src/assets/sass/components/_component-featured-card.scss
- Edit - main SCSS entry that imports the _component-*.scss files (add one @import line)
- Create (via Sitecore, then pulled) - template + rendering + Available Renderings update + test page + datasource

Verification checklist
- dotnet sitecore ser pull completes with new template/rendering files appearing under src/items/
- Dev Next.js site auto-reloads (it runs in container in dev mode - no restart needed)
- https://services.sxstarterer.localhost/test-featured-card renders the card
- Chrome DevTools MCP screenshot matches Figma node 50:1100 visually (layout, colors, typography, button shape)

Claude has written up a plan and is ready to execute. Would you like to proceed?
> 1. Yes, auto-accept edits
   2. Yes, manually approve edits
   3. No, refine with Ultraplan on Claude Code on the web
   4. Tell Claude what to change
      shift+tab to approve with this feedback

ctrl-g to edit in Notepad · ~\.claude\plans\structured-puzzling-hennessy.md
```

How it works: Sitecore items



```
• sitecore - item-service-create-item (MCP)(parentPath: "/sitecore/templates/Project/Verticals/Components/Featured Card/Data", itemName: "Link", templateId: "{455A3E98-A627-4B40-8035-E683A0331AC7}", data: {"Type": "General Link", "Title": "Link", "__Icon": "office/32x32/robot.png"}, options: {"database": "master"})
```

• Template done. Now create the rendering item. Let me first look at an existing Promo CTA rendering to understand field conventions.

```
• sitecore - item-service-get-item-by-path (MCP)(path: "/sitecore/layout/Renderings/Project/Verticals/Page Content/Promo CTA", options: {"database": "master", "includeStandardTemplateFields": true })
```

* **Creating Featured Card rendering...** (4m 53s · ↑ 6.3k tokens · thinking)

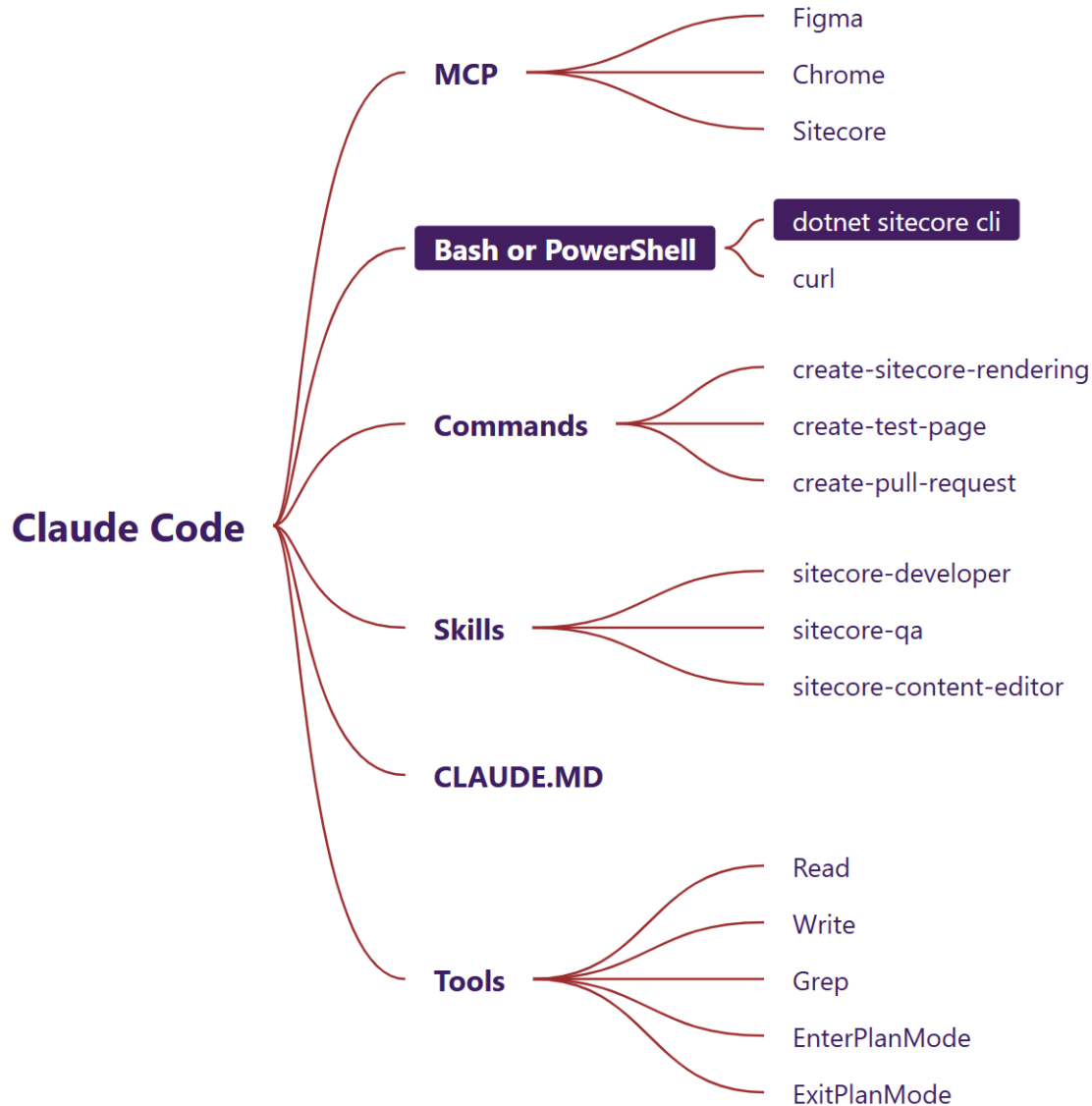
- ✓ ~~Create Featured Card template in Sitecore~~
 - **Create Featured Card rendering item**
 - Add rendering to Available Renderings Page Content
 - Pull Sitecore changes (first pull)
 - Create FeaturedCard React component and SCSS
 - Create test page with datasource and rendering
 - Pull Sitecore changes (second pull)
 - Verify test page matches Figma design

featured-card-rendering

> |

» accept edits on (shift+tab to cycle) · esc to interrupt · ctrl+t to h...

How it works: Pull serialization



```
· featured-card-rendering x + - □ x
```

```
97DD} | {316889FA-FDD4-42A6-A034-20112
4FE54E8} | {F3766B60-C8CE-49AF-9BFE-B2
CEE6E287DF} | {69E8A90F-59CA-4638-B53E
-BDCE65835230} | {5816F979-C8BB-46E9-8
EC8-513C6CAF41E6} | {3F98B12E-9693-47D
6-8507-18B054CCFB8E} | {7D1264B8-DBEA-
4DCD-8B2C-5DB934359EA9} | {09530D1F-14
1D-4885-8B3E-EBCA3EA5A23E} | {30ADE213
-D556-46D8-A97E-5638E9775E7F}"} ,
options: {"database":"master"})
```

- Now run the first `dotnet sitecore ser pull`.
- `Bash(dotnet sitecore ser pull 2>&1 | tail -60)`
 - L [master] [A] /sitecore/layout/Renderings/Project/Verticals/Page Content/Featured Card (30ade213-d556-46d8-a97e-5638e9775e7f)
 - [master] [A] /sitecore/templates/Project/Verticals/Components/Featu
 - ... +17 lines (ctrl+o to expand)
 - L (timeout 5m)
- Pull succeeded. Now create the frontend component files.

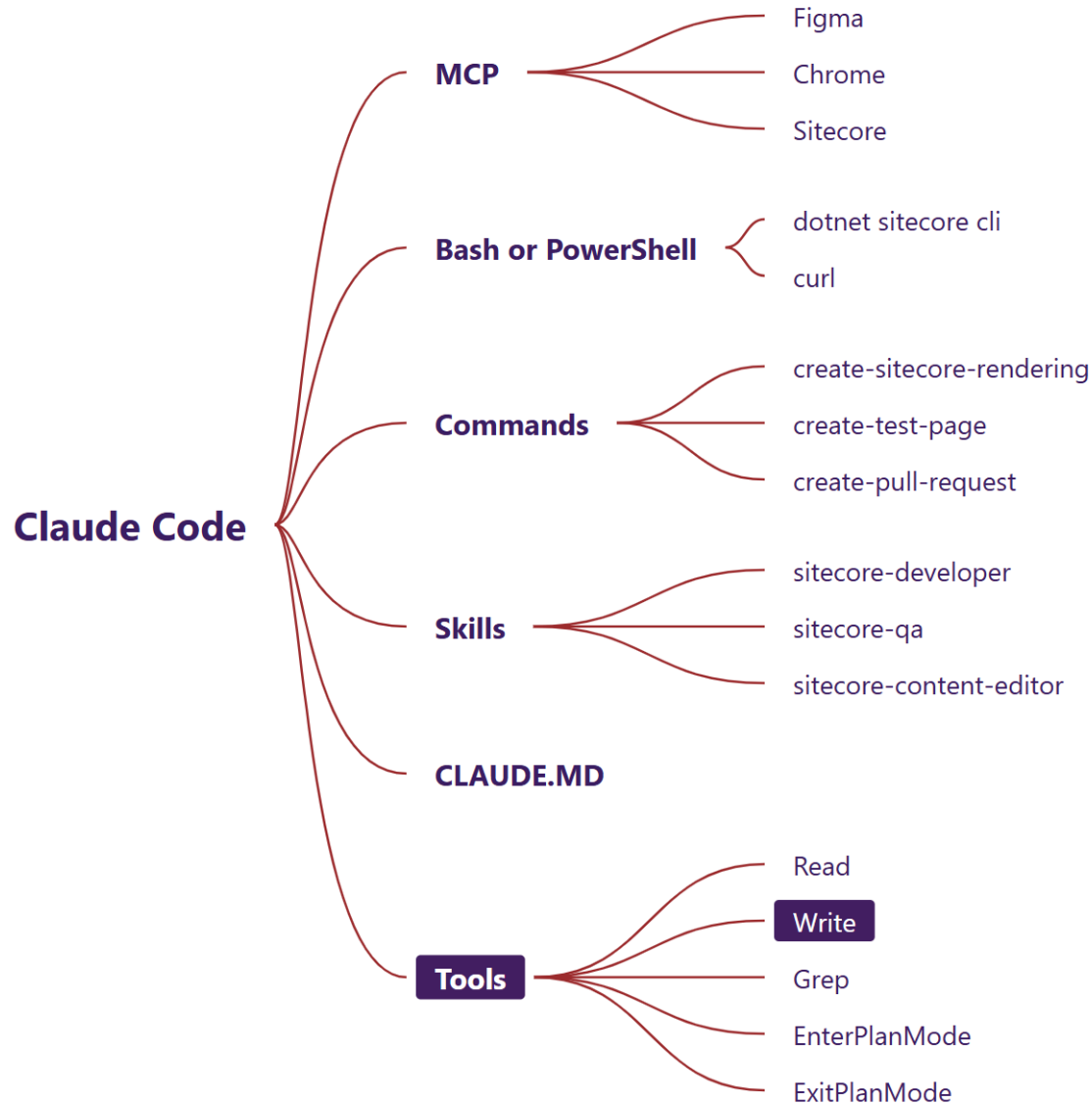
```
· Running sitecore ser pull... (5m 33s · ↓ 6.9k tokens)
L ✓ Create Featured Card template in Sitecore
  ✓ Create Featured Card rendering item
  ✓ Add rendering to Available Renderings Page Content
  ■ Pull Sitecore changes (first pull)
    □ Create FeaturedCard React component and SCSS
    □ Create test page with datasource and rendering
    □ Pull Sitecore changes (second pull)
    □ Verify test page matches Figma design
```

```
featured-card-rendering
```

```
> |
```

```
▸▸ accept edits on (shift+tab to cycle) · esc to interrupt · ctrl+t to h...
```

How it works: React component



```
· featured-card-rendering x + v - □ x
```

```
1 .featured-card {
2   padding-top: $component-margin;
3   padding-bottom: $component-margin;
4
5   @include respond-to(mobile-large) {
6     padding-left: $size-padding;
7     padding-right: $size-padding;
8   }
9
10  .featured-card-inner {
... +81 lines (ctrl+o to expand)
```

- Update(src\sxastarter-nextjs\src\assets\sass\components\index.scss)
 - Added 1 line
 - 35 @import '_component-project-details';
 - 36 @import '_component-project-list';
 - 37 @import '_component-promo-cta';
 - 38 +@import '_component-featured-card';
 - 39 @import 'component-accordion';
 - 40 @import 'common';
 - 41 @import 'container';

+ Creating FeaturedCard frontend files... (6m 4s · ↑ 8.1k tokens)

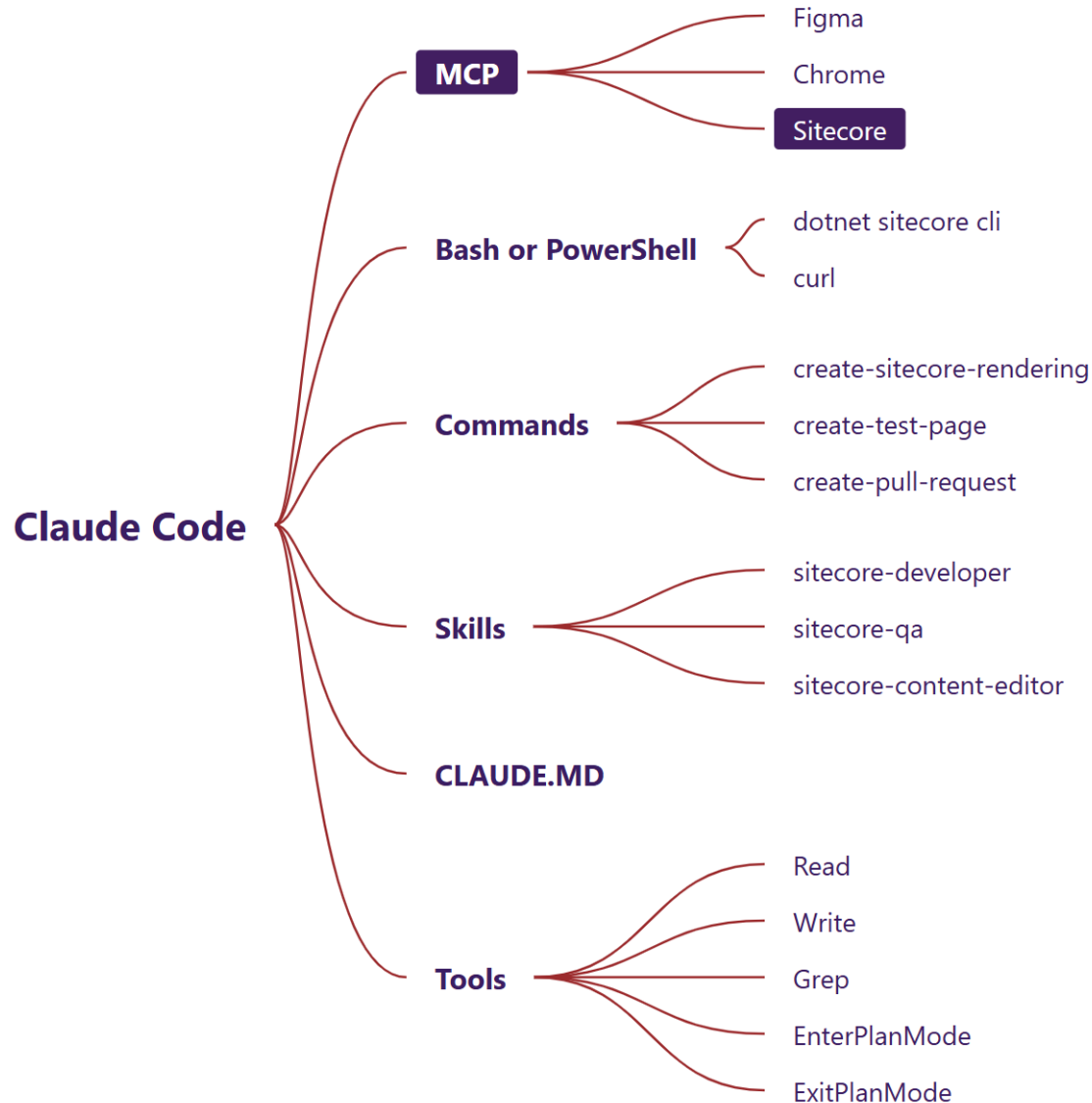
- ✓ Create Featured Card template in Sitecore
- ✓ Create Featured Card rendering item
- ✓ Add rendering to Available Renderings Page Content
- ✓ Pull Sitecore changes (first pull)
- Create FeaturedCard React component and SCSS
 - Create test page with datasource and rendering
 - Pull Sitecore changes (second pull)
 - Verify test page matches Figma design

```
featured-card-rendering
```

```
> |
```

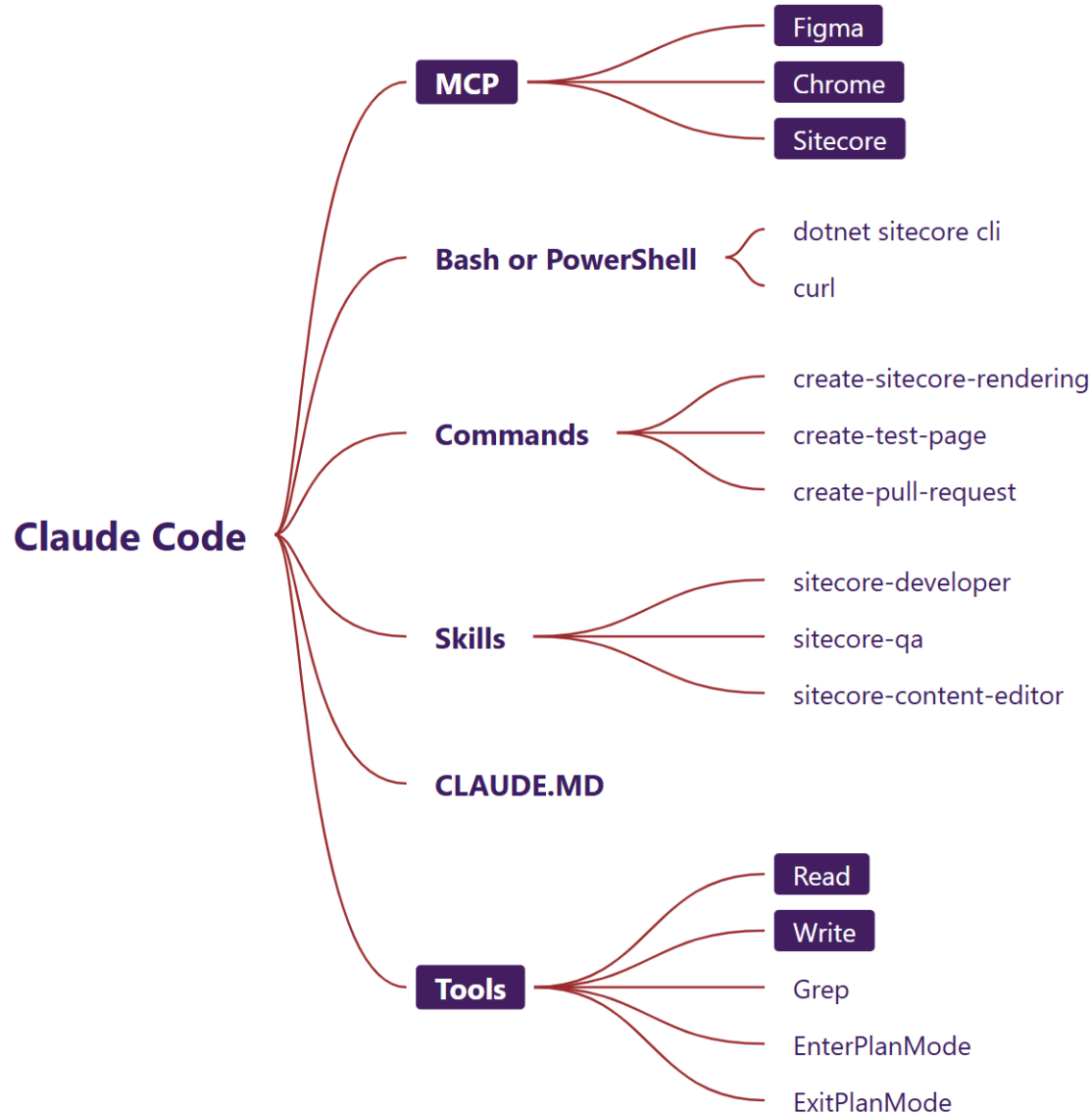
▸▸ accept edits on (shift+tab to cycle) · esc to interrupt · ctrl+t to h...

How it works: Test Page



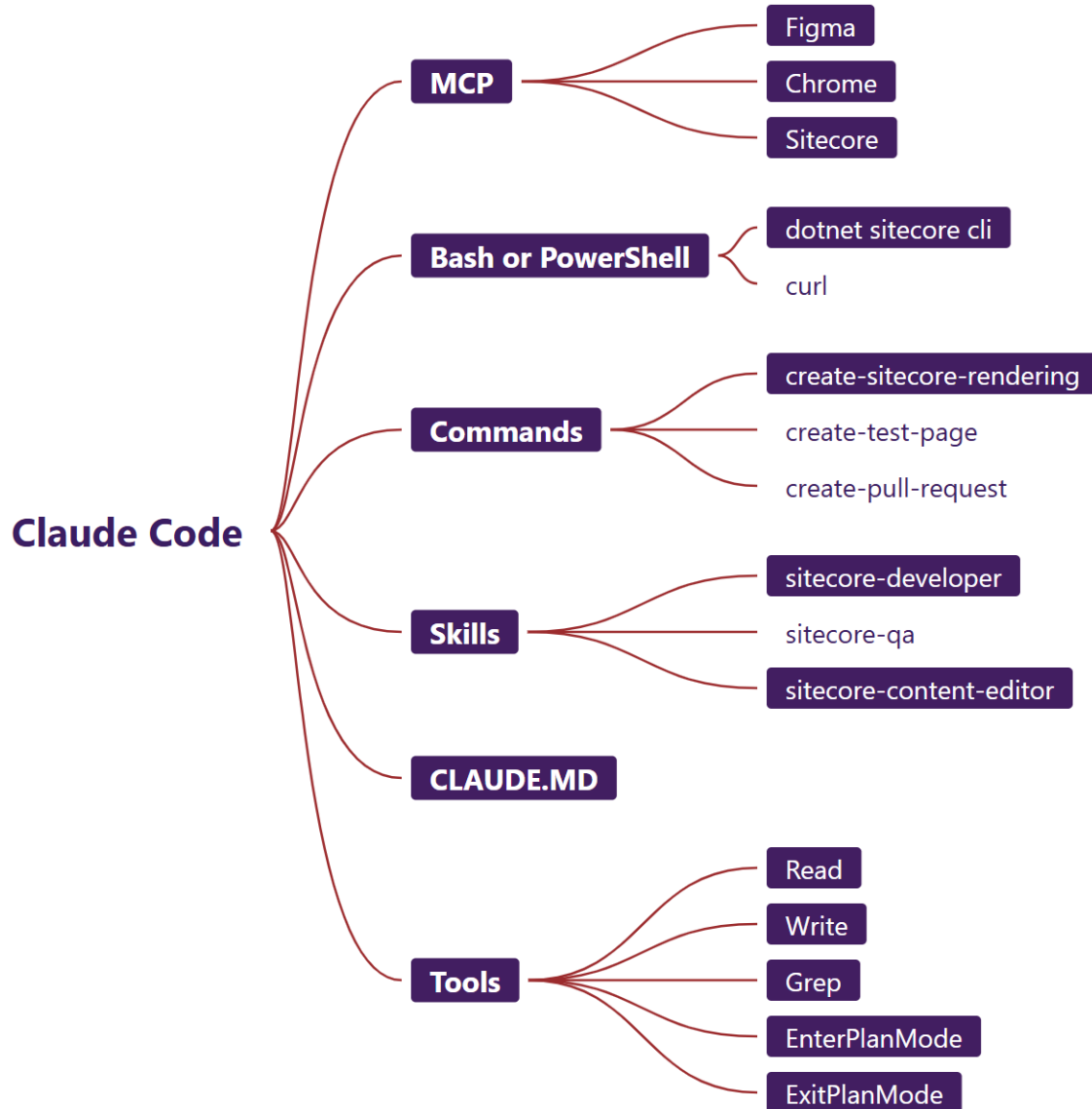
```
· featured-card-rendering × + - □ ×  
n": "office/32x32/robot.png"},  
options: {"database": "master"})  
• Now add the rendering to the test page.  
• sitecore - item-service-get-item-by-path (MCP)(path: "/sitecore/content/Verticals/Services/Home/test-featured-card", options: {"database": "master"})  
• sitecore - presentation-add-rendering-by-id (MCP)(itemId: "{AEABCD4F-E074-452D-9C4B-7A71D5BF4B45}", renderingId: "{30ADE213-D556-46D8-A97E-5638E9775E7F}", placeholder: "headless-main", dataSource: "local:/Data/Featured Card", finalLayout: true)  
* Creating test page... (7m 33s · ↑ 9.3k tokens · thinking)  
  L ✓ Create Featured Card template in Sitecore  
    ✓ Create Featured Card rendering item  
    ✓ Add rendering to Available Renderings Page Content  
    ✓ Pull Sitecore changes (first pull)  
    ✓ Create FeaturedCard React component and SCSS  
      ■ Create test page with datasource and rendering  
        □ Pull Sitecore changes (second pull)  
        □ Verify test page matches Figma design  
    > |  
    >> accept edits on (shift+tab to cycle) · esc to interrupt · ctrl+t to h...
```

How it works: Verification and fixes



```
• featured-card-rendering x + - □ x
42 + height: auto;
43 aspect-ratio: 1 / 1;
44 object-fit: cover;
45 }
46 }
47
48 .featured-card-content {
42 - flex: 1 1 auto;
49 + flex: 1 1 0;
50 + min-width: 0;
51 display: flex;
52 flex-direction: column;
53 align-items: flex-start;
• plugin:chrome-devtools-mcp:chrome-devtools - navigate_page (MCP)(type:
  "reload", ignoreCache: true, timeout: 60000)
  L Running...
* Verifying against Figma... (11m 20s · ↓ 10.6k tokens · thought for 10s)
  L ✓ Create Featured Card template in Sitecore
    ✓ Create Featured Card rendering item
    ✓ Add rendering to Available Renderings Page Content
    ✓ Pull Sitecore changes (first pull)
    ✓ Create FeaturedCard React component and SCSS
    ✓ Create test page with datasource and rendering
    ✓ Pull Sitecore changes (second pull)
    ■ Verify test page matches Figma design
  featured-card-rendering
> |
  >> accept edits on (shift+tab to cycle) · esc to interrupt · ctrl+t to h...
```

How it works: The result



featured-card-rendering x + - □ x

● Featured Card – done

Figma: node 50:1100 in Imported-Design
Live: <https://services.sxstarterer.localhost/test-featured-card>

Template	/sitecore/templates/Project/Verticals/Components/Featured Card – fields: Eyebrow, Title, Image, Link
Rendering	/sitecore/layout/Renderings/Project/Verticals/Page Content/Featured Card – componentName FeaturedCard, Datasource Resolver
Component	src/sxstarterer-nextjs/src/components/PageContent/FeaturedCard.tsx
Styles	src/sxstarterer-nextjs/src/assets/sass/components/_component-featured-card.scss (imported in components/index.scss)
Test page	/sitecore/content/Verticals/Services/Home/test-featured-card with datasource local:/Data/Featured Card on headless-main
Image	self_style_studio_at_greenwood_plaza_featured_card_8f0490 (1376x1376)

Rendered output verified against Figma via Chrome DevTools MCP – peach card, 632px square image left, eyebrow + title + pill CTA right.

featured-card-rendering

> |

» accept edits on (shift+tab to cycle)

Is Sitecore development solved by AI?

- Things that still require your attention
- Animations
- Corner cases
- Bugs in Pages (or Experience Editor)
- Orchestration
- Process improvement



Who should you be?



- [Silly Symphony The Tortoise and the Hare](#)
Disney, 1935
- [SUG Jaipur | Sitecore MCP in Action](#)
October 2025
- [Konabos webinar | AI That Actually Works](#)
[with Sitecore: Introducing the MCP Server](#)
September 2025
- [Sitecore MCP Server](#)
- [Figma MCP Server](#)
- [Chrome DevTools MCP Server](#)
- [EXDST](#)





Thank you to our sponsors!

Gold Sponsors



Technology Sponsor



Community Plus Sponsors



Catering Sponsor



Community Extra Sponsors



Community Sponsors



Lanyards Sponsor



Breakout Room Sponsors

