

Contents at a Glance

Introduction	1
Part 1: Getting Started with MCP	7
CHAPTER 1: Introducing the Model Context Protocol	9
CHAPTER 2: Setting Up Your MCP Environment	31
CHAPTER 3: Your First MCP Project	59
Part 2: Building with MCP Servers	89
CHAPTER 4: Working with Prebuilt MCP Servers	91
CHAPTER 5: Connecting Your Data	129
CHAPTER 6: Creating Your Own Custom MCP Client and Servers	159
CHAPTER 7: Security, Authentication, and Compliance	219
Part 3: Advanced MCP Applications	247
CHAPTER 8: Controlling Agent Behavior with System Prompts	249
CHAPTER 9: Building Production-Ready Solutions	267
CHAPTER 10: Performance, Scaling, and Reliability	313
CHAPTER 11: Integrating MCP into Existing Systems	333
Part 4: The Part of Tens	357
CHAPTER 12: Ten Essential MCP Servers	359
CHAPTER 13: Ten Common MCP Mistakes (And How to Avoid Them)	373
Index	395

Table of Contents

INTRODUCTION	1
About This Book	1
Foolish Assumptions	3
Icons Used in This Book	4
Beyond the Book	4
Where to Go from Here	5
PART 1: GETTING STARTED WITH MCP	7
CHAPTER 1: Introducing the Model Context Protocol	9
Why AI Needs External Data (And Why Integration Has Been Difficult)	10
What MCP Does, in Plain English	11
The Core MCP Components: Resources, Tools, and Prompts	15
Resources: Static data that feeds context	15
Tools: Functions AI can execute	17
Prompts: Reusable instruction templates	19
The complete picture	21
How MCP Is Being Implemented	21
Early adopters and reference implementations	21
Developer tools and IDE integration	22
Enterprise knowledge management	23
Data analysis and business intelligence	23
Customer support and service automation	24
Personal productivity and automation	24
The open-source ecosystem	25
Business Value: Why Your Organization Should Care	25
Reduced integration costs	26
Faster time to value	26
Enhanced security and compliance	27
Flexibility and future-proofing	27
Innovation enablement	28
Ecosystem leverage	28
CHAPTER 2: Setting Up Your MCP Environment	31
Choosing Your Path: Claude Desktop versus API Integration	32
The Claude Desktop path: MCP for humans	32
The API integration path: MCP for developers	33
Making the choice	34
Installing Claude Desktop and Enabling MCP	35
Step 1: Download Claude Desktop	36

Step 2: Install the application.	36
Step 3: Sign in.	37
Step 4: Enable MCP support	37
Step 5: Restart Claude Desktop.	39
Step 6: Verify that MCP is working	39
Testing with Prebuilt MCP Servers	41
MCP Server 1: The filesystem server — your files, Claude's knowledge.	41
MCP Server 2: The GitHub server repository access	46
Troubleshooting Your Claude Installation	51
The boot sequence	52
How a request works	53
Why this architecture matters	54
The configuration file's role	56
MCP protocol under the hood.	56
Resources versus tools versus prompts, revisited	57
CHAPTER 3: Your First MCP Project	59
Why This Project?	59
What You'll Build	60
Step 1: Set Up the Knowledge Base	61
Choose your documents	61
Organize your files	62
Set up the Python environment	62
Convert documents to a standard format.	64
Add metadata.	66
Step 2: Connect Claude to Your Documents	66
Build your first MCP server	67
Add search functionality.	70
Configure Claude Desktop	74
Step 3: Test and Refine Your Prompts	76
Start with simple questions	76
View the log files	84
PART 2: BUILDING WITH MCP SERVERS	89
CHAPTER 4: Working with Prebuilt MCP Servers	91
Discovering Available MCP Servers.	91
The official starting point: Anthropic's reference servers	92
Community hubs and aggregators	92
Searching with intent	93
Essential Servers for Business Users	94
The Filesystem server: Your AI's eyes on your files	94
The MySQL server: Talk to your database in plain English	94
The Fetch server: The whole web as context.	94

The YouTube Transcript server: Turn any video into text	95
The Puppeteer server: A browser your AI can drive.	95
Other servers worth installing early	95
Installing Popular MCP Servers	96
Tools you need.	98
Installing the Filesystem MCP server	99
Installing the MySQL MCP server	104
Installing the Fetch MCP server.	108
Installing the YouTube Transcript MCP server	110
Installing the Puppeteer MCP server	114
Combining Multiple Servers for Powerful Workflows	117
A simple multiserver workflow example	117
Video-to-action workflows	119
Managing the limits of multiserver workflows	121
Evaluating Server Quality and Trustworthiness	123
The evaluation checklist	123
The principle of least privilege.	124
Red flags to watch for	124
When to build versus buy.	125
When a prebuilt server is the right choice.	125
When building your own makes sense.	125
The cost of building.	126
A decision framework.	127
CHAPTER 5: Connecting Your Data	129
Database Integration Using SQLite.	130
Making your first database connection: SQLite	130
Setting up the SQLite MCP server	132
Trying out the SQLite MCP server	134
Filesystems and Document Repositories.	136
Setting up the Filesystem server	136
Working with CSV files	137
Working with PDF Files	140
Setting up the Markdown MCP server.	140
Summarizing a PDF document	142
Connecting to Gmail	143
Creating a new project	143
Enabling the Gmail API	143
Creating the credentials file.	149
Configuring the MCP config file.	152
Trying out the Gmail MCP server	152
Working with Cloud Storage: Google Drive	154
Setting up Google Drive	155
Setting up the Google Drive MCP server	156

CHAPTER 6:	Creating Your Own Custom MCP Client and Servers	159
	What Does an MCP Client Actually Do?	160
	The MCP handshake in plain English	162
	How MCP servers are deployed	162
	Setting up your environment.	164
	Connecting to an MCP Server Using stdio	165
	Implementing the MCP server using stdio.	165
	Implementing the MCP client using stdio	173
	Creating the MCP client with LLM tool calling.	175
	Connecting to an MCP Server Using SSE Transport	184
	Implementing the MCP server using SSE.	184
	Creating the MCP client with LLM tool calling.	193
	Connecting to Multiple MCP Servers	197
	Creating two MCP servers	198
	Creating the MCP client to use multiple MCP servers	210
CHAPTER 7:	Security, Authentication, and Compliance	219
	Analyzing the Vulnerabilities	220
	The two questions security has to answer	221
	Authentication: Locking the front door	221
	Encryption: Protecting the connection.	233
	Privacy and Safe Data Handling	241
	What server2 already does correctly	242
	Keeping secrets out of logs	242
	Controlling how long data lives in the cache.	243
	Compliance: What the Frameworks Require.	243
	General Data Protection Regulation (GDPR)	243
	Health Insurance Portability and Accountability Act (HIPAA)	244
	System and Organization Controls 2 (SOC 2)	245
	Security checklist for production deployments	245
	PART 3: ADVANCED MCP APPLICATIONS	247
CHAPTER 8:	Controlling Agent Behavior with System Prompts	249
	Dissecting the System Prompt.	250
	Customizing the MCP client with a system prompt	251
	Telling the model about its tools using a system prompt.	253
	Testing the system prompt	259
	Keeping the Agent Loop under Control	261
	What to Do When the Model Ignores Your Prompt	263
	Verify that the prompt is being sent.	264
	Make the language explicit	264

Add explicit do-not rules	264
Move the important rules to the top	265
Fall back to code	265
CHAPTER 9: Building Production-Ready Solutions	267
Case Study 1: Customer Support Automation with a Knowledge Base	268
What you're building	268
Architecture at a glance	269
The knowledge base schema	270
Installing the package	273
Building the MCP server	274
Starting the server	278
Wiring the server into Claude	278
Making the server production-ready	281
Caching the hot articles	282
Observability	283
Authentication	285
Testing the whole loop	285
Handling "I don't know" gracefully	286
What this pattern gives you	286
Case Study 2: Business Intelligence Dashboard with Real-Time Data	287
The hard part: Letting an LLM near your data	287
What you're building	288
The database schema	288
Installing the packages	291
Defining the metrics	291
Creating the server	293
Wiring the server into Claude	296
A real conversation	297
Production considerations specific to BI	302
The pattern, generalized	304
Patterns You Can Adapt to Your Needs	304
Pattern 1: Search, fetch, act	304
Pattern 2: Curated surface over a powerful backend	305
Pattern 3: Split by security profile	305
Pattern 4: Idempotency keys on writes	305
Pattern 5: Job IDs for long operations	306
Pattern 6: Catalogs as first-class tools	306
Pattern 7: Defense in depth around inputs	306
Pattern 8: Structured logging everywhere	307
Pattern 9: Read replicas for analytical tools	307
Pattern 10: A small system prompt beats a big one	307

From Prototype to Production: The Checklist	308
Section 1: Correctness	308
Section 2: Safety	308
Section 3: Observability	309
Section 4: Performance	309
Section 5: Deployment	309
Section 6: Authentication and network	310
Section 7: Data	310
Section 8: Documentation and ops	311
Section 9: The human layer	311
Section 10: Final sniff test	312
CHAPTER 10: Performance, Scaling, and Reliability	313
Optimizing MCP Server Performance	314
Where MCP servers usually become slow	314
Quick wins for faster servers	315
Caching Strategies That Work	316
What's safe to cache (and what isn't)	316
The three layers of caching	317
Cache invalidation: The hard part	318
Handling Multiple Concurrent Connections	319
Synchronous versus asynchronous: A quick refresher	319
Connection pooling: The unsung hero	320
Avoiding the pitfalls of shared state	321
Sizing your server for concurrency	322
Error Handling and Graceful Degradation	322
Retry with exponential back-off	323
Circuit breakers: Knowing when to give up	324
Returning useful errors	324
Monitoring, Logging, and Observability	325
The three pillars: Logs, metrics, and traces	325
Logging that doesn't make you cry	325
Metrics worth tracking	326
Setting up your first alert	327
Rate-Limiting and Resource Management	328
Where to apply rate limits	328
The token bucket: The algorithm you should know	328
Memory and resource caps	329
High Availability Architectures	329
Running multiple instances	330
Surviving when the data center goes dark	331
Deployment without downtime	331
Backup and recovery	332

CHAPTER 11: Integrating MCP into Existing Systems	333
Assessment: Is MCP Right for Your Use Case?	334
The core MCP value proposition	334
The 5-question fitness test	334
Common situations where MCP may not be the best fit	336
Integration Patterns and Architectures	336
The sidecar server	337
The API gateway integration	337
The embedded integration	338
The shared service (MCP hub)	339
Deciding between stdio and SSE transport	340
Working with Legacy Systems	340
Evaluating the legacy integration ladder	340
Wrapping legacy databases	341
Dealing with legacy data formats	342
When there's no API and no direct database access	343
Conquering legacy authentication challenges	343
Change Management and Team Adoption	344
Accommodating stakeholder perspectives	344
Emulating the champion model	345
Addressing the job security question	345
Training and enablement	346
Establishing governance early	346
Migration Strategies for Existing AI Solutions	347
Determining whether to migrate	347
Examining the strangler fig pattern	348
Migrating hard-coded API integrations	348
Migrating RAG pipelines	349
Data migration considerations	350
Measuring success and ROI	350
Key Takeaways	354
 PART 4: THE PART OF TENS	 357
CHAPTER 12: Ten Essential MCP Servers	359
Official Anthropic Servers	360
Filesystem server	360
Fetch server	360
Memory server	361
Database and Data Warehouse Connectors	361
PostgreSQL and SQLite	361
BigQuery	362
Snowflake	362

Productivity Tool Integrations	362
Google Workspace (Gmail, Calendar, Drive)	363
Slack	363
Notion and Obsidian	363
Development and DevOps Utilities	364
GitHub	364
GitLab	364
Docker and Kubernetes	365
Terminal/Shell	365
Analytics and Monitoring Solutions	365
Grafana	366
Datadog	366
Amplitude and Mixpanel	366
Specialized Industry Servers	366
Legal research (Westlaw and Casetext)	367
Financial data (Bloomberg and Alpha Vantage)	367
Healthcare and life sciences	367
Community-Built Game Changers	367
Browser automation (Playwright and Puppeteer)	367
Home automation (Home Assistant)	368
Music and media (Spotify)	368
Experimental Servers to Watch	368
AI-to-AI communication servers	368
Real-time data streams	369
Multimodal servers	369
How to Evaluate New Servers	369
Security first	369
Maintenance and activity	370
Documentation quality	370
Community adoption	370
Contributing to the MCP Ecosystem	370
Reporting issues and improving docs	370
Building your first server	371
Publishing and sharing	371
The Big Picture	371

CHAPTER 13: Ten Common MCP Mistakes (And How to Avoid Them)	373
Skipping Security in Development	374
The minimum you should always have	375
Authentication for MCP specifically	375
Overcomplicating Your First Project	376
Ignoring Error Handling	377
Poor Resource and Tool Design	379

Inadequate Testing Coverage	380
Hard-Coding Configuration Values	383
What belongs in configuration, not code	383
The .env.example trick	384
Neglecting Documentation	385
The README you should have	385
A README skeleton you can copy	386
Failing to Plan for Scale	387
Components to get right early on (because they're painful to change)	387
Components to defer (because they're cheap to add later)	388
The 1 client today, 100 tomorrow test	389
Forgetting about the User Experience	389
Trying to Do Everything with MCP	391
When MCP is the correct answer	391
When MCP probably isn't	392
A quick decision guide	393
Your 20-Minute Self-Audit	394
INDEX	395

