

# Table of Contents

<b>Preface.....</b>	<b>xv</b>
<b>1. The Enterprise AI Conundrum.....</b>	<b>1</b>
The AI Landscape: A Technical Perspective All the Way to GenAI	3
Machine Learning: The Foundation of Today's AI	4
Deep Learning: A Powerful Tool in the AI Arsenal	4
Generative AI: The Future of Content Generation	5
Open Source Models and Training Data	8
Why Open Source Is an Important Driver for GenAI	8
The Hidden Cost of Bad Data: Understanding Model Behavior Through Training Inputs	9
Adding Company-Specific Data to LLMs	10
Explainable and Transparent AI Decisions	10
Ethical and Sustainability Considerations	11
The Lifecycle of LLMs and Ways to Influence Their Behavior	12
MLOps Versus DevOps (and the Rise of AIOps and GenAIOps)	13
Conclusion	15
<b>2. The New Types of Applications.....</b>	<b>17</b>
Understanding Large Language Models	18
Key Elements of a Large Language Model	19
Deployment of Models	25
Choosing the Right LLM for Your Application	33
Model Type	33
Model Size and Efficiency	34
Deployment Approaches	34
Supported Precision and Hardware Optimization	34
Ethical Considerations and Bias	35

Community and Documentation Support	35
Closed Versus Open Source	35
Example Categorization	36
Foundation Models or Expert Models: Where Are We Headed?	38
Using Supporting Technologies	39
Embedding Models and Vector Databases	39
Caching and Performance Optimization	40
AI Agent Frameworks	40
Model Context Protocol	41
API Integration	41
Model Security, Compliance, and Access Control	42
Conclusion	44
<b>3. Prompts for Developers: Why Prompts Matter in AI-Infused Applications.....</b>	<b>45</b>
Types of Prompts	45
User Prompts: Direct Input from the User	45
System Prompts: Instructions That Guide Model Behavior	46
Contextual Prompts: Prepopulated or Dynamically Generated Inputs	46
Principles of Writing Effective Prompts	46
Prompting Techniques	47
Zero-Shot Prompting: Asking Without Context	47
Few-Shot Prompting: Providing Examples to Guide Responses	47
Chain-of-Thought Prompting: Encouraging Step-by-Step Reasoning	48
Self-Consistency: Improving Accuracy by Generating Multiple Responses	48
Instruction Prompting: Directing the Model Explicitly	49
Retrieval-Augmented Generation: Enhancing Prompts with External Data	49
Advanced Strategies	49
Constructing Dynamic Prompts: Combining Static and Generated Inputs	49
Using Prompt Chaining to Maintain Context	50
Using Guardrails and Validations for Safer Outputs	50
Leveraging APIs for Prompt Customization	51
Optimizing for Performance Versus Cost	51
Debugging Prompts: Troubleshooting Poor Responses	51
Tool Use and Function Calling	52
Context Engineering as the New Prompt Engineering	53
Designing Memory and Storage for Context	54
Fast Access with In-Memory Caches	54
Hot Memory for Short-Term Context	54
Vector Databases for Long-Term Semantic Memory	54
Cold Storage for Archival Data and Large Repositories	55

Combining Storage Tiers for Effective Context Delivery	55
Conclusion	55
<b>4. AI Architectures for Applications</b>	<b>57</b>
Beyond Traditional Architectures: Why AI-Infused Systems	
Require a New Approach	57
Overview of Core Architectural Pillars: A Roadmap for the Chapter	59
Application Components	60
Queries and Data: Managing Application Inputs	61
The AI Gateway: Managing Inputs and Outputs	63
Context and Memory	66
Interaction and Transport: Using Tools and Agents	69
Discovery and Access Control	72
Model Serving	73
The Data Preparation Pipeline	76
Observability and Monitoring: The End-to-End AI Stack	78
Conclusion	80
<b>5. Embedding Vectors, Vector Stores, and Running Models Locally</b>	<b>83</b>
Embedding Vectors and Their Role	83
Why Are Embeddings Needed?	84
Structure of an Embedding Vector	85
Measuring Similarity: Cosine Similarity and Distance	85
Common Embedding Models	88
How Are Embeddings Used in AI Applications?	91
Other Similarity Methods	93
Uncommon Uses of Embedding Vectors	95
Vector Stores and Querying Mechanisms	97
How Vector Databases Store and Retrieve Embeddings	97
Examples of Common Vector Stores	98
Retrieval-Augmented Generation	100
Indexing or Generating Vector Embeddings at Scale	102
Why Run Models Locally?	104
Ollama: Local Inferencing with a Simple Interface	106
Podman Desktop: Using Containerized Environments for AI Workloads	109
Jlama: Java-Native Model Inferencing for JVM-Based Applications	120
Comparing Local Inferencing Methods	123
Using OpenAI's REST API	125
Overview of OpenAI's Models and Endpoints	126
Generating Embeddings with OpenAI's API	129
Conclusion	132

<b>6. Inference APIs.....</b>	<b>133</b>
What Is an Inference API?	133
Benefits of an Inference API	135
Examples of Inference APIs	135
Deploying Inference Models in Java	139
Inferencing Models with DJL	140
Looking Under the Hood	147
Inferencing Models with gRPC	148
Conclusion	154
<b>7. Accessing the Inference Model with Java.....</b>	<b>155</b>
Connecting to an Inference API with Quarkus	155
The Architecture	156
The Fraud Inference API	156
The Quarkus Project	157
The REST Client Interface	157
The REST Resource	158
Testing the Example	159
Connecting to an Inference API with Spring Boot WebClient	160
Adding WebClient Dependency	160
Using the WebClient	160
Connecting to the Inference API with the Quarkus gRPC Client	161
Adding gRPC Dependencies	161
Implementing the gRPC Client	162
Conclusion	164
<b>8. LangChain4j.....</b>	<b>167</b>
What Is LangChain4j?	167
Unified APIs	168
Prompt Templates	170
Structured Outputs	172
Memory	174
Data Augmentation	176
Tools	179
High-Level API	181
LangChain4j with Plain Java	185
Extracting Information from Unstructured Text	185
Performing Text Classification	187
Generating Images and Descriptions	190
Spring Boot Integration	192
Adding Spring Boot Dependencies	193
Defining the AI Service	194

Creating a REST Controller	195
Quarkus Integration	196
Quarkus Dependencies	197
Frontend	198
The AI Service	199
WebSocket	201
Optical Character Recognition	203
Tools	205
Dependencies	207
Rides Persistence	207
Waiting Times Service	209
AI Service	210
REST Endpoint	211
Dynamic Tooling	213
Final Notes About Tooling	218
Memory	219
Dependencies	222
Changes to Code	222
Conclusion	223
<b>9. Vector Embeddings and Stores</b>	<b>225</b>
Calculating Vector Embeddings	225
Vector Embeddings Using DJL	226
Vector Embeddings Using In-Process LangChain4j	228
Vector Embeddings Using Remote Models with LangChain4j	232
Text Classifier	233
Embedding Text-Classification Dependencies	234
Providing Examples and Categorizing Inputs	234
Text Clustering	236
Adding Text Clustering Dependencies	237
Reading Headline News	237
Calculating the Vector Embedding	238
Clustering News	239
Summarizing News Headlines	241
Semantic Search	243
Adding Semantic Search Dependencies	244
Importing Movies	246
Querying for Similarities	251
Semantic Cache	254
RAG	257
Ingestion	258
Retrieval	263

Reranking	267
Query Router	269
Ingestion Splitting Window	273
Filtering Results	277
Conclusion	280
<b>10. LangGraph4j.....</b>	<b>283</b>
Understanding Graphs in LangGraph4j	284
Nodes	284
Edges	285
State	286
Using LangGraph4j	287
Defining a State	288
Defining a Node	289
Defining a Graph	289
Adding Conditional Edges	291
Appending Values	293
Using LangChain4j with LangGraph4j	294
Routing Agents	295
Human Interaction with LangGraph4j	299
Advanced RAG Schema with Self-Reflection	310
Exploring Additional Features	312
Subgraphs	312
Parallel Execution	313
Time Travel	314
Conclusion	315
<b>11. Image Processing.....</b>	<b>317</b>
OpenCV	319
Initializing the Library	320
Loading and Saving Images	320
Performing Basic Transformations	322
Overlaying Elements	325
Image Processing	330
Reading Barcodes and QR Codes	343
Stream Processing	346
Processing Videos	346
Processing Webcam Images	347
OpenCV and Java	348
OCR	350
Conclusion	353

<b>12. Advanced Topics in AI Java Development.....</b>	<b>355</b>
Streaming	356
Streaming with a Low-Level API	356
Streaming with AI Services	357
Using LangChain4j and Streaming Integrations	358
Guardrails	360
Input Guardrail	361
Output Guardrail	363
Guardrail Use Cases	365
Model Context Protocol	367
MCP Architecture	368
MCP Client with Java	371
MCP Client with Quarkus	375
MCP Server with Quarkus	379
Key Benefits of MCP	388
Next Steps	389
<b>Index.....</b>	<b>391</b>

## Beyond Prototypes: Building Resilient AI-Infused Applications with Java

When we started discussing the early draft release of this book, we quickly received a lot of feedback. One comment stuck with me that went along the lines of "AI is an AI expert these days, and unless you have 10 years of experience in AI, you should not write a book like this." That strong warning sound like impostor syndrome in all of us. But it also gave us an opportunity to reiterate why we wanted to write this book and share our view on enterprise application development in these times of AI with you.

We have seen a lot of companies starting to infuse AI into existing applications. Companies are eager to quickly use AI features to enhance user experience, optimize and automate workflows, and speed up systems. However, the reality of this push