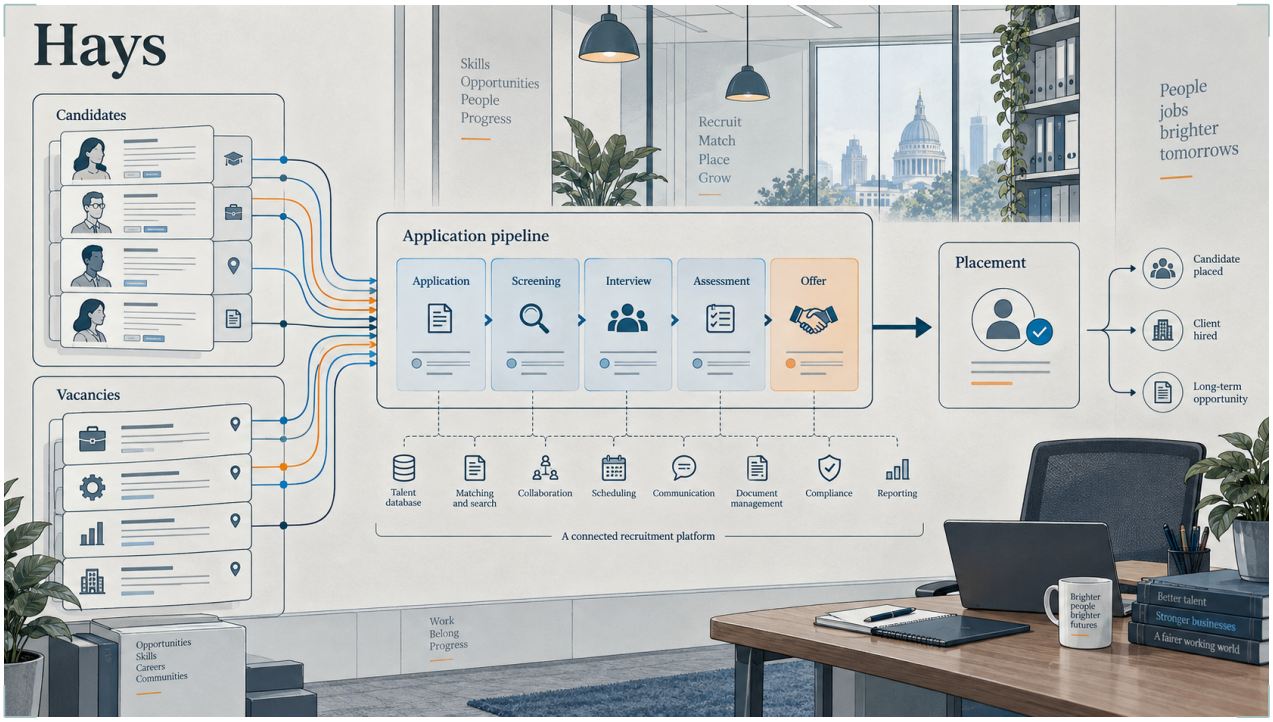
Recruitment Platforms, Built for Continuity

My Hays experience across Matchbox and 3 Story Software



Gerry Walter

Senior Applied AI Engineer

Senior Full Stack Developer | March 2022 - April 2023

Software I worked on. Engineering knowledge I bring.
A personal portfolio of my contribution and experience.

Cover: AI-generated conceptual editorial illustration, not an application screenshot.

START HERE

Introduction and Contents

A compact case study for technical and stakeholder presentations.

At Hays AG Germany, I worked as a Senior Full Stack Developer. My application experience included Matchbox Neo, the wider Matchbox environment and 3 Story Software. Working remotely from Thailand, my Hays role combined frontend development, backend services and enterprise integration in recruiting software.

My work covered candidate, client, vacancy, application and placement workflows. I contributed to system design, maintainable components and service boundaries, automated testing and CI/CD. I also investigated defects across the frontend, backend and integration layers while collaborating with German product, recruiting, business and engineering stakeholders.

Contents

01 **What I Developed at Hays** 3

02 **How I Understood the Recruiting Domain** 4

03 **My Full-Stack Development Practice** 5

04 **Matchbox and 3 Story Software** 6

05 **The Breadth of My Contribution** 7

06 **How I Worked on Quality and Delivery** 8

07 **Glossary and References** 9

08 **Closing Perspective** 10

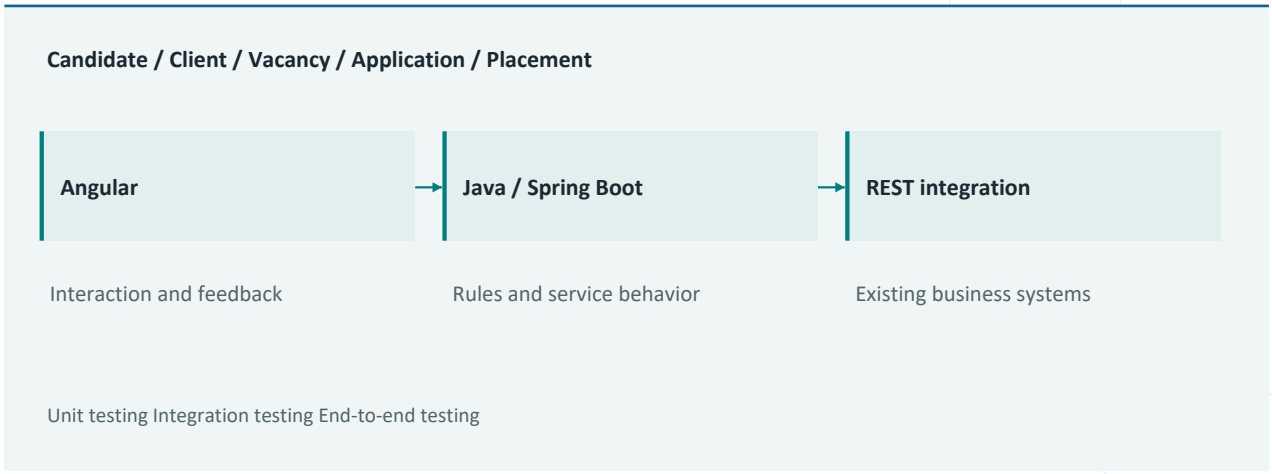
My engineering approach

I bring a full-stack view to enterprise software: understand the business process, translate it into application behavior and follow that behavior through every layer. At Hays, that meant connecting the recruiting experience with the services and integrations underneath it, while maintaining continuity in an established application environment.

CHAPTER 01

What I Developed at Hays

Full-stack implementation for the Matchbox recruitment platform.



01 My work combined Angular, Java/Spring Boot and REST across five recruiting workflow domains.

Frontend and backend delivery

I developed Matchbox Neo functionality using Angular on the frontend and Java/Spring Boot on the backend. My work connected user-facing behavior with the services required to support it. I contributed to implementation across both layers, with maintainable components, clear service boundaries and enterprise integration forming part of the development responsibility.

Five connected recruiting workflows

I worked on candidate, client, vacancy, application and placement workflows. These were related parts of an internal recruiting platform, not isolated screens. Understanding how information moved between them was important to translating requirements into application functionality and API interactions that supported the recruiting process.

Working with the business

I collaborated with German product, recruiting, business and engineering stakeholders while working remotely from Thailand. My role included translating operational requirements into components and service behavior. That required understanding the recruiting task behind a request, discussing its implications and connecting the business perspective to implementation decisions.

Responsibility beyond new features

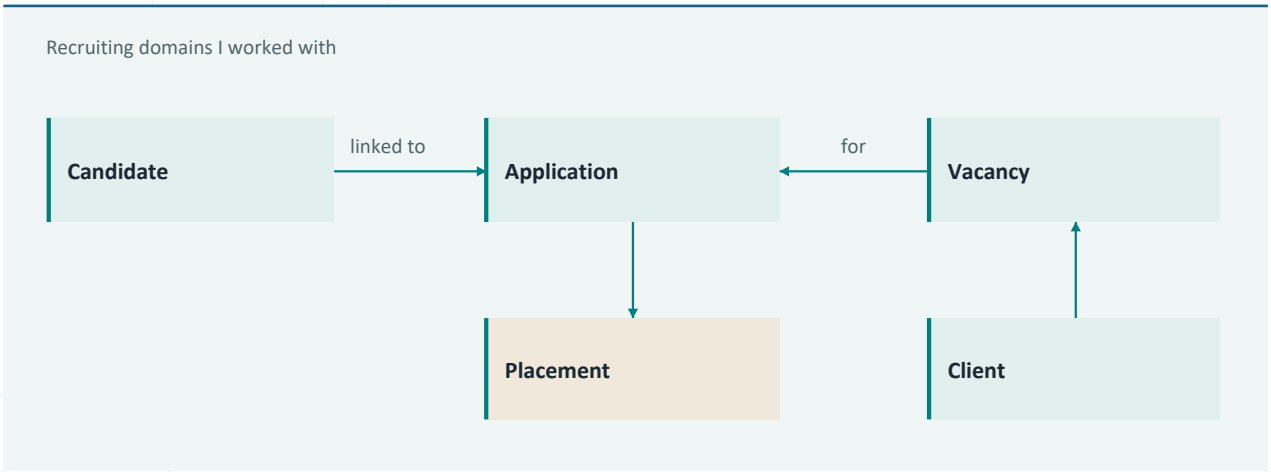
I also worked on automated tests, regression prevention, documentation and repeatable CI/CD. Defect investigation crossed frontend, backend and integration layers. My contribution therefore included both developing capabilities and helping maintain a stable application as those capabilities changed.

Angular / Java / Spring Boot / REST / Recruitment workflows | References: p. 9

CHAPTER 02

How I Understood the Recruiting Domain

Connecting business meaning with maintainable application design.



02 The five workflow domains I worked with. This relationship diagram summarizes their business meaning rather than an internal database schema.

Workflows before components

My system-design work started with recruiting requirements and how they related to the application. Candidate information, a client requirement, an open vacancy and a placement serve different purposes. Understanding those distinctions helped me translate business needs into frontend components, service responsibilities and API interactions.

The application connects the process

An application links a candidate to a particular opportunity, while a placement represents a later business outcome. This distinction is part of the domain knowledge I bring to recruitment software. It helps explain why a change must be considered in the context of the workflow, rather than only as an update to a form or record.

Maintainable frontend design

I contributed Angular frontend functionality and components. My work required aligning the user interaction with the underlying service behavior. For me, frontend engineering is not only about rendering a view: it is about helping the user understand the state of their work and complete the next action through a coherent interaction.

Service boundaries with purpose

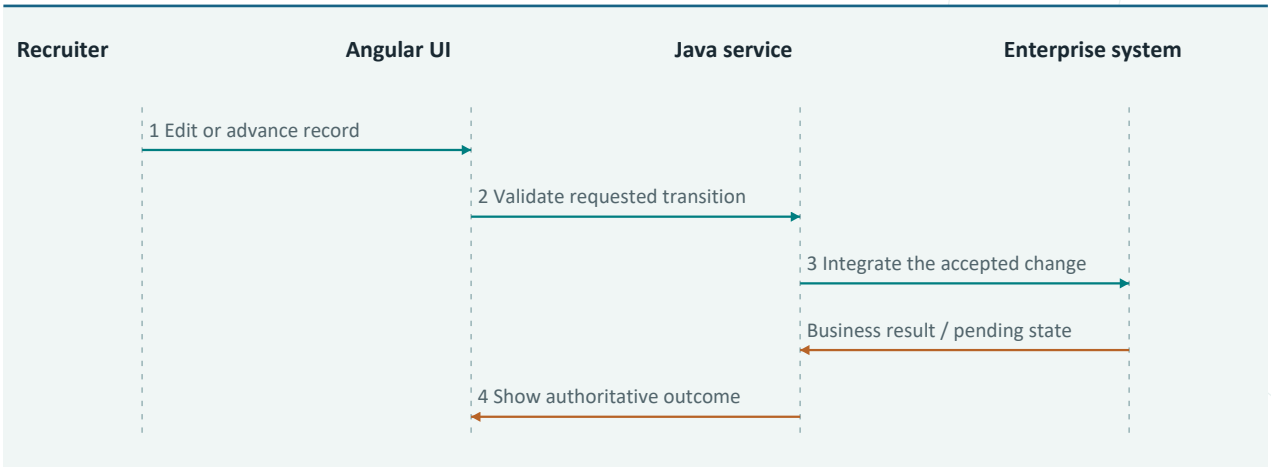
I contributed Java/Spring Boot backend services and REST interactions. Maintainable architecture was part of that work. I connect service boundaries to business responsibility, so that a frontend change, an integration or a defect investigation can be understood in terms of the recruiting operation it affects.

Domain modeling / Component design / Service boundaries | References: p. 9

CHAPTER 03

My Full-Stack Development Practice

Following a recruiting action from the interface to the connected business system.



03 A simplified interaction across the layers I worked on: recruiter, Angular frontend, Java service and enterprise integration.

From requirement to user interaction

I translated operational recruiting requirements into application components and interactions. Working on Angular functionality meant understanding the task the recruiter was trying to complete and the information the service needed. I connected those concerns instead of treating the interface as independent of backend behavior.

From interaction to service behavior

On the backend, I developed Java/Spring Boot functionality and REST APIs. My work included service boundaries and interactions supporting the recruitment workflows. This gave me practical experience following a feature through the request, the business behavior and the response that the frontend had to handle.

From service to enterprise integration

I worked on REST integrations with established recruitment and business systems. This extended the responsibility beyond the local application. Understanding connected systems and their interactions was important both when implementing functionality and when investigating a problem that appeared in the recruiting interface.

From behavior to a regression test

I contributed unit, integration and end-to-end testing. These levels allowed me to address local functionality, connected behavior and the complete user path. My development practice links a fix or feature to the behavior it is intended to deliver, with regression work helping the team preserve that behavior through later changes.

Frontend development / Backend services / REST integration / Testing | References: p. 9

CHAPTER 04

Matchbox and 3 Story Software

My application experience across recruitment and workforce management.

My Hays application experience: Matchbox and 3 Story Software.

04 My Hays application landscape included Matchbox and 3SS. The platforms are presented as distinct systems, not one shared architecture.

Matchbox Neo and Legacy context

My Hays work centered on Matchbox Neo and included the wider Matchbox Legacy environment. Working with established software required attention to existing behavior as well as new requirements. I approached changes with the recruiting workflow in mind, because the business needed continuity while the application continued to evolve.

My work included 3SS

I also worked on 3 Story Software at Hays. This adds workforce management to my application experience alongside Matchbox's recruitment workflows. 3SS is Hays' vendor management system, supporting the coordination of external resources and suppliers. My Hays experience therefore spans both the recruiting application context and work on the 3SS product.

Understand the cause across layers

I investigated defects spanning the frontend, backend and integrations. That work requires moving beyond the place where a problem becomes visible. A recruiting screen may show the symptom, while the underlying issue sits in service behavior or a connected system. My full-stack background lets me follow that path and contribute to the correction.

Connect people and implementation

I worked with product, recruiting, business and engineering stakeholders in Germany. That collaboration helped connect operational requirements with technical decisions. I bring the ability to discuss a workflow with the people using it and then work through its implementation details with the engineering team.

Matchbox / Maintainability / Cross-layer investigation / Collaboration | References: p. 9

CHAPTER 05

The Breadth of My Contribution

A personal view of the engineering responsibilities I covered at Hays.

5
recruiting workflow domains

3
connected technical layers

3
automated testing levels

Frontend	**Angular**	User-facing functionality
Backend / integration	**Java / Spring Boot**	Services and REST interactions
Quality and delivery	**Unit / integration / E2E**	Regression and repeatable CI/CD

05 These counts summarize my work areas, not business-performance benchmarks: five workflow domains, three test levels and three connected technical layers.

Five workflow domains

My implementation scope included candidates, clients, vacancies, applications and placements. Working across these domains gave me a view of the recruiting process as a connected application. I could relate an individual component or service change to the wider workflow rather than treating it as a standalone development task.

Three connected technical layers

I worked across Angular frontend functionality, Java/Spring Boot backend services and REST enterprise integration. This breadth mattered during both feature development and defect investigation. I could follow behavior between the user interaction, service implementation and connected system instead of stopping at a single layer.

Three levels of automated testing

My quality work included unit, integration and end-to-end tests, together with regression prevention and CI/CD. These responsibilities complemented implementation. They helped make changes repeatable and supported the continuity expected of an internal recruiting platform.

The value I brought to the team

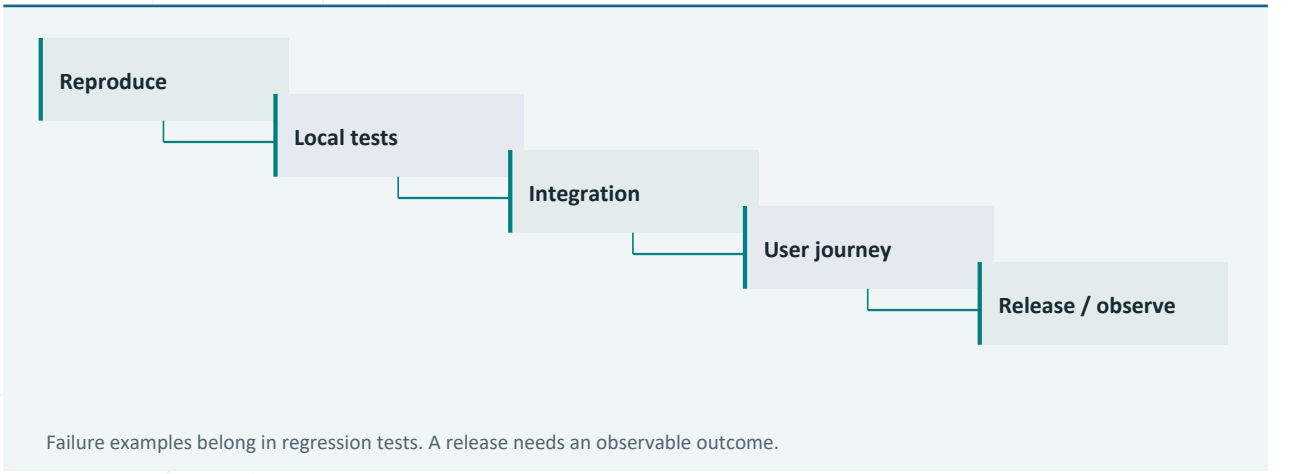
I combined hands-on full-stack development with domain understanding and collaboration across product, business and engineering. My contribution included system design, implementation, investigation and delivery. That combination is the value of this engagement in my portfolio: practical ownership of development work across an established enterprise application.

Personal scope: workflow domains / technical layers / test levels | References: p. 9

CHAPTER 06

How I Worked on Quality and Delivery

Development, regression prevention and cross-layer problem solving.



Failure examples belong in regression tests. A release needs an observable outcome.

06 My delivery responsibilities combined implementation, automated tests, defect investigation, documentation and repeatable CI/CD.

Testing at different levels

I contributed unit tests for focused behavior, integration tests for connected functionality and end-to-end tests for complete user workflows. Working with all three levels helped me consider a change from more than one perspective. It also connected development to the recruiting task that the application needed to support.

Defect investigation and regression

I investigated issues across Angular, backend services and enterprise integrations. My work included regression prevention, so a correction was part of maintaining the application rather than simply removing a visible symptom. This experience strengthened my ability to reason across system boundaries and connect a technical issue with its user-facing effect.

Documentation and repeatable delivery

I contributed technical documentation and repeatable CI/CD processes alongside feature development. These responsibilities matter in a shared codebase: implementation needs to remain understandable, changes need to be testable and the team needs a consistent delivery path. I worked on that ongoing engineering process as part of the role.

My software engineering foundation

The Hays engagement shows my full-stack range and my experience with enterprise application development. I bring that foundation to applied AI work today: understanding business entities, implementing services, integrating systems and testing complete workflows. New capabilities become useful when they fit the application and the people who rely on it.

Unit / Integration / End-to-end / Regression / CI/CD | References: p. 9

REFERENCE

Glossary and References

Technical terms, keyword page references and supporting project context.

Supporting References

[CV] Gerry Walter: career and project history

Personal role, dates, technology stack and contribution details supplied by the author.

[R] Project research and contextual references

Supporting research supplied by the author, 12 September 2026. Distinguishes application history and external business context.

[H1] Hays: 3 Story Software

Official product background for the VMS application. Personal participation confirmed by Gerry Walter; public product features are not an attribution of individual feature ownership.

www.hays.de

Glossary / Keyword Index

3SS / pp. 2, 6

3 Story Software: the Hays vendor management application I also worked on.

Angular / pp. 3-5

Frontend framework I used for recruitment-platform functionality.

Application / pp. 3-5

The recruiting workflow linking a candidate to an opportunity.

CI/CD / pp. 7-8

Repeatable integration and delivery supported by automated checks.

Integration testing / pp. 5, 8

Checking that application layers and connected services work together.

REST / pp. 4-6

API interaction style used in my backend and enterprise integration work.

Spring Boot / pp. 3-5

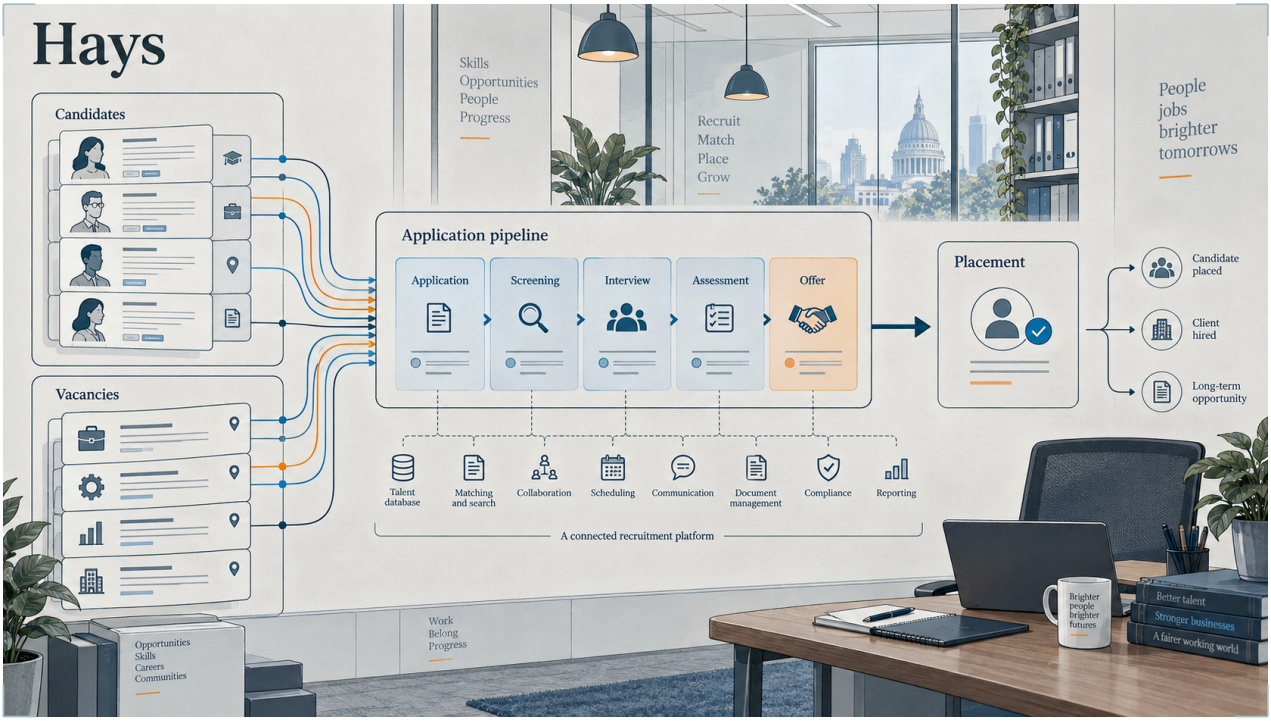
Java framework I used for backend services.

Visual provenance

Cover and closing artwork: newly AI-generated conceptual illustrations. Technical diagrams: original, editable explanatory drawings. Research excerpts: cropped from the user-supplied PDF where indicated. None is presented as a private application screenshot.

CLOSING PERSPECTIVE

I build enterprise software around the people using it.



My Hays experience included Matchbox and 3 Story Software. Across the role, I contributed Angular interfaces, Java/Spring Boot services, REST integration, automated testing and delivery. My Matchbox work supported candidates, vacancies, applications and placements, with maintainability and cross-layer problem solving as part of the responsibility.

I bring this enterprise development experience to my work as a Senior Applied AI Engineer. Understanding business entities, application state and system integration is what lets me connect new capabilities to software people already rely on. My Hays experience is a practical foundation for that work.

Gerry Walter

Senior Applied AI Engineer

Hays / Personal engineering portfolio / September 2026