Got it - you've got a solid engineering foundation already, which is exactly what a Technical Solutions Architect role builds on. The key now is to pivot from "how do I code this?" to "how do I design this for scale, cost, and business value?". Let's map out a practical path for you:

Step 1: Reframe Your Experience

Instead of presenting yourself as "developer," highlight the architectural decisions you've already made:

- Designed scalable enterprise apps with Spring Boot and Node.js.
- Balanced SQL vs NoSQL trade-offs for performance and flexibility.
- Implemented Redis caching to reduce latency and improve throughput.

These are already architect-level contributions - you just need to frame them that way.

Step 2: Build Architect-Specific Skills

You've mastered backend stacks. Now expand into:

- Cloud platforms: Pick one (AWS, Azure, GCP) and go deep.
- System design patterns: Microservices, event-driven, CQRS, API gateways.
- Security & compliance: Identity management, encryption, GDPR basics.
- Cost awareness: How architecture choices affect cloud bills.

Step 3: Certifications That Signal Readiness

- AWS Certified Solutions Architect  Associate/Professional
- Microsoft Certified: Azure Solutions Architect Expert
- Google Professional Cloud Architect

These force you to think like an architect and give recruiters confidence.

Step 4: Networking & Job Search

- LinkedIn presence: Share architecture diagrams, write short posts about trade-offs (SQL vs NoSQL, monolith vs microservices).

- Communities: Cloud meetups, architecture forums, open-source contributions.

- Target roles: "Solutions Architect," "Technical Architect," "Cloud Architect."

- Companies: Cloud providers (AWS, Azure, GCP), consulting firms (Accenture, Deloitte), product companies (Salesforce, Atlassian, ServiceNow).

Developer vs. Architect Focus

| Area | Developer (You now) | Architect (Target) |
|---|---|---|
| Coding | Deep expertise | High-level guidance |
| Databases | Query optimization | Choosing right DB |
| Caching | Implement Redis | Designing strategy |
| Cloud | Limited exposure | Multi-cloud design |
| Communication | Team collaboration | Stakeholder-facing |
| Business Alignment | Feature delivery | Translating needs |

Next Actions

1. Pick a cloud platform and start certification prep.

2. Practice system design problems weekly (mock interviews, whiteboard diagrams).

3. Rewrite your CV to emphasize architecture decisions.

4. Share insights publicly - it builds credibility.

Would you like me to draft a sample CV summary that positions you as a future Technical Solutions Architect? That way you can see how to reframe your developer experience into architect language.