

# Canonical Written Interview

## Engineering Experience

### Question 1

What kinds of software projects have you worked on before? Which operating systems, development environments, languages, databases?

### Answer 1

Throughout my career, I have worked on a wide variety of software projects, including big data processing, reporting, web app solutions, and analytics platforms. I have experience with multiple operating systems, development environments, languages, and databases. Here's a summary of some of the technologies and tools I have worked with:

#### Operating Systems

Linux (various distributions), Windows, macOS

#### Development Environments

Visual Studio Code, PyCharm, IntelliJ IDEA, Eclipse

#### Languages

Python, JavaScript, SQL, GraphQL, Java

#### Databases

PostgreSQL, Aurora PostgreSQL, Oracle, Netezza, DynamoDB

#### Frameworks and Libraries

FastAPI, Tartiflette, Apollo GraphQL, Pandas, Spark (PySpark)

#### Big Data and Cloud Technologies

Hadoop (Cloudera), Apache Airflow, AWS (Lambda, ECS, EMR, S3, Glue, Stepfunctions, etc.)

#### Other Tools and Technologies

Docker, Kubernetes (AWS EKS), Kafka (AWS MSK), GitLab CI/CD, Github, Grafana, Prometheus, AlertManager, Sentry

## **Summary**

This list is not exhaustive, but it gives you an idea of the range of technologies and tools I have worked with in my career. I have hands-on experience developing software for various industries and have learned successful technology leadership strategies by working on different projects, technologies, programming languages, countries, and organizations.

## **Question 2**

Would you describe yourself as a high quality coder? Why?

## **Answer 2**

Yes, I would describe myself as a high-quality coder. In my opinion supported by experience, code which is hard to maintain is expensive for an organization. Therefore, high quality code is not only resilient but also easy to maintain. Easy to maintain code focuses on declaring to the reader what it does via a clean structure and naming considerations. Below are some points to support my claim.

### **Wide-ranging Experience**

I have worked on a number of projects, technologies, and programming languages over the course of my nearly 20 years of software development expertise. I now have a thorough understanding of coding standards, design patterns, and best practices thanks to this experience.

### **SOLID and Clean Code Advocate**

My understanding of code structure theory and hands on experience make me well suited to advocate for principles such as Open/Close, Dependency Injection, and Single Responsibility. In general following these principles along with layered architecture and declarative code reduces the cost of maintaining the code over it's lifetime.

### **Ongoing Development**

I'm committed to staying current on trends and developments in software development. I routinely take part in online forums, read industry publications, and experiment with new technologies to sharpen my coding skills and expand my knowledge.

### **Excellent Problem-solving Skills**

Writing high-quality code requires the ability to comprehend complex problems and provide workable solutions. My professional experience has helped me hone my problem-solving skills, which have enabled me to take on challenging assignments and deliver superior results.

### **Team Player**

I understand that high-quality code is often the result of collaboration and teamwork. I have experience leading and working with diverse teams of engineers, and I am committed to fostering a positive, inclusive environment where everyone can contribute their best work.

### **Focus on Quality**

I recognize the importance of thorough testing and quality assurance in producing high-quality code. Most of my code comes with tests to ensure the expected behavior is achieved. I have experience implementing testing frameworks, conducting code reviews, and working closely with QA teams to ensure that the software I develop is reliable, secure, and performs as expected.

## **Question 3**

Would you describe yourself as an architect of resilient software? If so, why, and in which sorts of applications?

### **Answer 4**

Yes, I would describe myself as an architect of resilient software. My experience in designing and implementing robust, scalable, and maintainable systems across various domains, such as big data processing, analytics platforms, and web applications, demonstrates my ability to create resilient software. See examples and project details below.

#### **Microservices architecture**

Breaking down applications into smaller, independent services that can be developed, deployed, and scaled independently.

#### **Event-driven architecture**

Decoupling components through asynchronous communication, allowing for better fault tolerance and scalability.

#### **Distributed systems**

Ensuring high availability and fault tolerance by distributing data and processing across multiple nodes.

## **Example Projects**

### **Global Panel Tools**

I implemented a microservices architecture, allowing for independent development and deployment of various components, improving scalability and fault tolerance.

### **International Media Measurement Processing Modernization**

I designed a distributed system using Apache Airflow, Docker, and AWS services, ensuring high availability and fault tolerance while handling large volumes of data.

## **Summary**

My experience in designing and implementing resilient software architectures across various projects showcases my ability to create robust and maintainable systems.