**People find their way into Systems Engineering from a wide range of backgrounds, and for different reasons. What attracts you personally to Systems Engineering?**

As my journey through computer science and software engineering has progressed, I've increasingly gravitated towards roles that require not just technical acumen but also a deep understanding of system-wide impacts and customer-centric thinking. My attraction to Systems Engineering is rooted in its holistic approach — it’s about designing and implementing complex yet beautiful systems that are both functional and aesthetically pleasing. This field involves integrating various subsystems into a cohesive whole that serves real-world needs, a process I find truly enjoyable and immensely fulfilling.

My experience at eBay, where I developed a dashboard to monitor cluster statuses, and at Kaizntree, where I co-founded and led the technology direction, exemplifies my ability to understand and implement solutions that consider broader systems. In these roles, I gained a deep understanding of how different software components interact within larger ecosystems, aligning perfectly with the core principles of Systems Engineering.

Furthermore, my academic background in computer science and data science has equipped me with a solid foundation in the technical aspects of systems. Concurrently, my involvement in projects like Parkrowd and Rate My Professor has honed my skills in translating complex requirements into actionable solutions.

My passion for Systems Engineering is further intensified by the artistic and creative aspects of designing and building complex systems. I see a unique beauty in the way various components, each with its own function and characteristics, come together to form an intricate and efficient whole. This synergy, akin to a well-orchestrated symphony, transforms Systems Engineering from a mere technical pursuit into an art form. Engaging in this discipline brings me true joy and satisfaction.

**What is the last technical skill that you taught yourself? How did you learn it? And why did you want to learn that particular skill?**

The most recent technical skill I taught myself is Docker and the principles of containerization. This learning journey began during my tenure at Kaizntree, where we were developing a comprehensive management platform for small businesses. As the project progressed, we faced a significant challenge: ensuring a consistent development and deployment environment for all team members. It was crucial to have a setup where every developer could work in an environment that mirrored our production settings, thus minimizing 'it works on my machine' issues.

To address this, I turned my attention to Docker. I started by exploring a range of online resources, such as official Docker documentation, tutorials, forums, and blogs written by containerization experts. I set up a personal testing ground using my computer, which allowed me to experiment with creating Docker containers, understanding Dockerfiles, and managing containerized applications.

What made this self-learning journey truly effective was its direct application to our work at Kaizntree. I began by implementing Docker for smaller, more manageable parts of our project. This involved containerizing individual applications and integrating them into our existing workflow. Gradually, as I became more adept, I expanded this to include more complex segments of our platform.

The decision to learn Docker stemmed from our need to maintain a consistent development and deployment environment across our team. This uniformity was crucial for streamlining our development process and reducing discrepancies between different working environments. By implementing Docker, we achieved a level of efficiency and consistency that significantly improved our development lifecycle.

This experience not only enhanced our project at Kaizntree but also reinforced my commitment to continuous learning and adapting to new technologies. It highlighted the importance of keeping pace with evolving tech trends, especially those that directly impact and improve our development practices.

**Give an example of a time you had to answer a technical “why?” question, similar to the examples in the job description above. Who was your audience and how did you provide the rationale for “why?**

One memorable instance of addressing a technical 'why?' question occurred during my time as a Software Engineering Intern with eBay's Cloud Infrastructure Team. The audience for this question was a mix of senior engineers and project managers. The question arose during the development of the Average-Time-to-Business (ATB) Dashboard, a tool I proposed and implemented using Redux, React, and Django.

The specific 'why?' question I encountered was: 'Why should we adopt this ATB Dashboard for monitoring cluster statuses?' This question was crucial as it challenged the necessity and effectiveness of the dashboard within our existing infrastructure.

To provide a comprehensive rationale, I first approached this question by collecting data that demonstrated the current inefficiencies in our crisis response system. I analyzed historical incident response times and the impact of service downtimes on business operations. Then, I showcased how the ATB Dashboard could reduce response times and boost service availability, using a prototype to demonstrate its real-time monitoring capabilities and alert system.

Furthermore, I conducted a comparative analysis with existing tools to highlight the additional functionalities and improvements that the ATB Dashboard offered. This included aspects like its intuitive UI, faster data processing, and more accurate alerts.

In addition to these technical details, I addressed the broader business impact. I articulated how enhancing service availability from 99.14% to 99.8% could significantly affect customer satisfaction and trust, ultimately contributing to the company's bottom line.

By combining data analysis, practical demonstrations, and a clear link to business benefits, I successfully provided a well-rounded answer to the 'why?' question. This not only satisfied the audience's curiosity but also garnered support for implementing the ATB Dashboard across eBay's cloud infrastructure.

This experience underscored the importance of not just understanding the technical aspects of a project, but also being able to articulate its impact and necessity in a broader business context.