



Senior Software Engine Architect C++ – Creative Engine

Full-time

Remote

Engineering

Who we are

We are that super-groundbreaking technology and design company you've probably never heard of...yet! Our flagship products include Software Development Kits for photo- and video-editing that are used to build custom applications by small and huge enterprises all over the globe.

We enjoy building great products that excite, empower, and connect people so they can make the world more creative and expressive. We are still a small team of 30 people but all together we have many years of experience creating products. Engineers, designers, sales, finance, hr, – we have all experiences onboard enabling us to iterate and build things quickly.

Where we work

As a remote-first company, we work where we work best. May it be from home, in an office, or a co-working space. img.ly was founded in Bochum, Germany and we have a small office there to hang out once in a while. We often work miles apart but still feel close. Aside from working together we hang out once in a while remotely and at least once per year all together somewhere on the globe.

What you will be working on

You will be working on our core technologies. Thus developing our in-house creative engine UBQ. The UBQ engine is a portable cloud-native runtime for creative editing needs implemented in C++ and used in all our new products.

The core of the engine focuses on graphics, animation, and print quality output. Its primary focus is the development of creative tools such as Photo-, Video-Editing Applications as well as Desktop Publishing Applications.

How we work

The best way to build a good product is to use your product. Be your customer. Our designers are using our tools and so we are in constant dialog with them to plan and implement our features.

We strive to develop in small pull requests with substantial reviews. Formatting, linting, tests will be done by our hardworking bots so we can focus on the important parts: Ensuring high code quality and sharing knowledge.

We do not work in sprints nor marathons. We need room to breathe without getting bored, so we work in cycles of 6 weeks divided into phases of tinkering and getting things done. Once it is done, we know it will not be perfect. Nothing is perfect from scratch. Continuous refinement and making time to pay off technical debt aren't just empty promises, but always part of the next cycle.

What can you bring to the table?

This isn't your next code monkey job where you transform coffee into code. Since we operate highly autonomously within our cycles you need to plan out new features by collaborating with your peers and

automatically through our eyes, you need to plan out how features by collaborating with your peers and design an architecture that will have the right balance between getting it done and not being a pain in the ass in the future.

Of course, it is essential to know your tools. We all know that **C++** can get messy quickly, and this is not a code-golf challenge. So we need to take special care to write understandable, clean, and yet performant code. You need to know how to do this.

It is alright to miss stuff here and there. What is important for every team member is to happily share your knowledge and ideas for improvement in code reviews and architectural discussions. And this is not a one-way street. Being able to receive and properly evaluate constructive feedback is equally important for us.

These are the basics. From there you learn the rest very quickly. But what would make you shine brightly is a broad understanding of **modern renderer pipelines** and **game engines**. Is **data driven design** and an **entity component system** your state of mind? Have you worked on **compute-, render and frame-graphs**? Have you maybe even tinkered with **vector or text rendering on the GPU or CPU**? Or deep-dived into writing your own game engine in C++ or Rust for fun? If you nod in agreement to any of this, we need to talk to you about this.

As a cherry on top, if you love building a positive relationship with your fellow engineers in the industry and not only in your company by joining meetups, going to or speaking at conferences, sharing your wisdom in your blog, or writing and maintaining open-source software, then we definitely need to talk to you.

How to apply

Get in touch

We don't need a novel as a cover letter. Sample projects are worth more than a thousand words. If you have a reference project feel free to include it with your CV.

Let's chat

The first interview with our people manager Kathi will be done with the purpose to get to know each other on a people level. This is, to set the frame for everything there is to come.

Meet the team

Our second interview will be a 20-minute remote coffee with the tech team to get to know each other with the intention to have a general discussion. We will tell you more about us and answer your questions, but obviously, we want to learn more about you as well. What are your interests? What have you been working on lately and why?

However, it will not be a technical interview, so you will not solve or program anything, but we might ask you some technical questions about your past projects according to your resume.

Test the waters

In the third interview, we like to get a glimpse and see how you work and communicate. The best way for us to do this is to see you in action. You will get a small feature to implement. Nothing fancy, nothing real, but enough to see how you approach a problem. You can work on this on your own, but we will jump in from time to time where you tell us what you have done, if you are stuck and why, and how you want to tackle the next issues. Basically, it will be a little bit like Pair Programming.

That's it!

[Apply now](#)

