Seattle, WA joe@schaffernation.com

joseph schaffer

Software engineer in Seattle. At Amazon, I'm the tech lead of a compiled programming language for authoring safe, secure user interfaces. I lead its technical direction across the type system, compiler, and tooling, and grew the team from 4 to 10. I work across front-end paradigms, from server-rendered templates to reactive client apps, and I care a lot about how an interface feels to use. Outside work, I tinker, build bikes, and make small, designy apps I'd want to use myself.

WORK

Amazon

Seattle

L4 2014 L5 2016 L6 2019

2018

8 yrs

Tech lead, Amazon's language for building user interfaces

TEAM	10 engineers
SCALE	part of ~80% of amazon.com pages
SECURITY	Amazon's preferred XSS defense
FOCUS	language design, type systems, compilers, interop with other languages
STACK	Kotlin compiler; JVM and JavaScript targets
TOOLING	language server, MCP server

I joined in 2018, when the team was four engineers, and I've been its tech lead since 2019. Its first version offered a safer, more runtime-performant way to template HTML. Today, because of my work partnering with the three server and micro-front-end platform teams, it's a full UI-focused language that is involved in rendering part of about 80% of amazon.com pages, and the team is 10 people. It's statically typed and compiles to the JVM and JavaScript, for server-rendered pages, the browser, and React Native.

- Set the language's direction with the team each year, toward making it the easiest way for people and AI agents alike to write UI that's right the first time, and kept platform, customer-experience, and feature teams aligned behind it.
- Designed and led the compiler's support for reactivity along two separate paths: direct interop with React, and compiling templates to a fine-grained reactive framework that updates only the markup that depends on changed state and hydrates only a page's interactive parts. Four engineers built the framework; I led its compiler integration. Reactive templates now run across thousands of Amazon shopping experiences.
- Led a ground-up rewrite of the type system: sound, and built to work alongside the languages it interoperates with. Its error messages, inspired by Rust's, point to the exact spot in the code and say how to fix it. Signatures are optional. Leave one off and the compiler infers it, as templating has always allowed, though writing your own is the recommended practice.
- Cross-site scripting protection lives in the compiler, which knows where each value lands in the page and escapes it for that spot. Amazon's security team recommends it as the first line of defense for server-rendered HTML.
- Grew the tooling around the compiler: a language server for editors and, more recently, an MCP server, so AI agents get the same errors and hints engineers do.
- Base design decisions on data: research into how other languages handle the same problems, feedback from the engineers who use it, and the compiler speed metrics we track to catch regressions.
- Mentored engineers through seven promotions, five to L5 and two to L6.

2016
2 yrs

A front-end build system and asset pipeline

SCALE thousands of teams across Amazon shopping
STACK JVM core (Java, Kotlin), Ruby, JavaScript

I was a core contributor to the front-end build tooling behind Amazon's retail pages (similar to Grunt, Gulp, or webpack) and the system that ships their assets to the CDN. I introduced switching tools that let teams declare variants without building every combination, and designed the plugin framework teams extend it with. Every page builds as a matrix of variants (device × locale × language × A/B test), each sliced to carry only what it needs, which was the key to fast pages before HTTP/2. Each new dimension multiplied the build, and the switching tools kept builds fast as the matrix grew.

2014
2 yrs

A front-end framework and component library

TEAM 10-20 engineers

The foundation of Amazon's retail pages. I worked across it — the carousel on mobile, tablet, and desktop, sweeping refactors, a TV navigation interface for the web — and designed how new versions were released to product teams.

ON THE SIDE

SchafferNation

Seattle

Founder

2026
<1 yr

A one-person studio for small, designy apps and tools

Trailer Trash	All the latest trailers, sorted by what's coming, playing, or gone from theaters. trailertrash.app	
Vintage NYC Guessr	Guess where historical photos of New York were taken.	schaffernation.com/nycguessr
Seattle Sports Today	Is Seattle playing today? Plan your commute accordingly.	seattlesportstoday.com

SCHOOL

University of Pennsylvania

Philadelphia

2010
4 yrs

DEGREE B.S.E.
MAJOR Computer Science
MINOR Mechanical Engineering
YEARS 2010 - 2014

REPERTOIRE

DAILY	Kotlin, TypeScript, JavaScript, React, Next.js, Node, HTML & CSS, Tailwind
ALSO	Swift, SwiftUI, React Native, Svelte, Vue, Rust, Java, Ruby, Objective-C, Go
UNDERNEATH	compilers, parsers, type checkers, the Language Server Protocol, MCP
LATELY	LLM APIs, agents, Claude Code
FONDLY	Haskell, OCaml, Coq
MAKING	SolidWorks, Illustrator, the machine shop (mill, lathe, CNC)
OTHERWISE	Typography, saxophone, bicycling, design of every kind, and how things work

UPDATED NOVEMBER 2026