

joseph schaffer

Software engineer in Seattle. At Amazon, I've worked on an internal programming language for user interfaces since 2018 and led it since 2019; it's used across Amazon shopping. I know front-end work across its 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	10-12 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. It started as a safer way to template HTML. Today it's a full UI language that renders part of about 80% of amazon.com pages, and the team is 10-12 people. It's statically typed and compiles to the JVM and JavaScript, for server-rendered pages, the browser, and React Native.
- Designed and built its reactivity, which works two ways: templates interoperate directly with React, or run on the language's own reactive framework, which updates only the markup that depends on changed state and hydrates only a page's interactive parts. I led a team of five to ship it, and it now runs across thousands of Amazon shopping experiences.
 - Led a ground-up rewrite of the type system. It should feel familiar to anyone who writes TypeScript, the types are sound, and nobody writes a template's signature by hand.
 - 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.
 - 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.
 - Base design decisions on data: research into how other languages handle the same problems, feedback from the engineers who use it, and the compile-time metrics we track to catch regressions.
 - Mentored engineers through seven promotions, five to L5 and two to L6.

2016

II

2 yrs

A front-end build system and asset pipeline

SCALE	thousands of teams across Amazon shopping	The front-end build tooling behind Amazon's retail pages (similar to Grunt, Gulp, or webpack), paired with the system that ships their assets to the CDN. Every page builds as a matrix of variants — device × locale × language × A/B test — with each combination sliced to carry only what it needs (the key to fast pages before HTTP/2), so each new dimension multiplies the build. As a core contributor, I introduced switching tools that let teams declare variants without building every combination, keeping builds fast as the matrix grows, and designed the plugin framework teams extend it with.
STACK	JVM core (Java, Kotlin), Ruby, JavaScript	

2014

II

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

Founder

Seattle

2026

I

<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

IIII

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