

joseph schaffer

Software engineer in Seattle. At Amazon, I lead the design of a compiled programming language for safe, secure user interfaces, used across Amazon shopping. Outside work, I tinker, build bikes, and make small, designy apps I'd want to use myself.

WORK

Amazon

Seattle

SDE I 2014 SDE II 2016 Senior SDE 2019

2018 A programming language for user interfaces

■■■■■■■■ 8 yrs	ROLE	Tech lead	I lead an internal programming language for building user interfaces: statically typed, compiled to both the JVM and JavaScript, and grown from a safe templating language for HTML into a full UI language that lives alongside TypeScript. I own its governance — syntax, type system, compiler architecture, and releases. Its tooling shares the compiler's understanding and its precise error messages: a language server for editors, an MCP server for AI agents, and a plugin architecture other teams build their own static analysis on.
	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	I lead its direction with the team: a north star for where the language is going, a plan each year that moves it closer, and reprioritizing through the year as we learn. The north star is an approachable language for front-end work that catches problems in the compiler, early, so engineers and AI agents alike can build UI quickly and safely, made for Amazon's polyglot stacks. Each step has built toward it: a typed templating language that met Amazon's UI developers where they were, with type safety and security built in; a language server that brought the compiler's understanding into every editor; and interop with TypeScript's type definitions, so it works alongside the code teams already have. We sequence and cut work so the long-term pieces land, and ground new features in analysis of the wider industry and in what the engineers who use it tell us; I turn that analysis into the case for each change in design reviews. We built metrics on compile time that catch performance regressions before they ship, and I use them in reviews to hold the line on compile speed.
	STACK	Kotlin compiler; JVM and JavaScript targets	
	TOOLING	language server, MCP server	
			Its guarantees are built in. The compiler knows where every value lands in the markup, and the runtime that writes the final bytes escapes each one for exactly that place, so templates are safe from cross-site scripting by construction. Amazon's security organization names it the first-choice defense against cross-site scripting in server-rendered HTML, ranked ahead of manual output encoding. Escaping only what each context needs keeps rendering fast. The type system is structural and sound, in the spirit of TypeScript, and I led its ground-up redesign. Template authors write no type signatures: the compiler infers each template's signature from how its parameters are used.
			I designed and built its reactivity, the model React.js, Svelte, and SwiftUI share: a template declares its view as a function of state, and when state changes, only the markup that depends on it updates. Reactivity is its own layer of the type system, so the compiler knows which values are state and where they reach, and each template compiles to a server render plus a client bundle that hydrates only its reactive regions. I led five engineers in delivering it, and it runs across thousands of Amazon shopping experiences.
			Along the way, I've mentored engineers to seven promotions: five to SDE II, two to senior.

2016

II 2 yrs

A front-end build system and asset pipeline

SCALE thousands of teams across Amazon shopping

STACK JVM core (Java, Kotlin), Ruby, JavaScript

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.

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

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, React, Next.js, Node, HTML & CSS, Tailwind

ALSO Rust, Swift, Objective-C, Java, Go

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