

FAN YOU

Software Engineer

(516) · 359 · 8605 ◊ contact@fyou.me

EDUCATION

Carnegie Mellon University, Entertainment Technology Center (ETC)

May 2016

Master of Entertainment Technology

Course: Distributed Systems, Algorithms and Advanced Data Structures, Building Virtual World.

New York Institute of Technology

May 2014

B.S. in Computer Science

Course: Operating Systems, Computer Architecture, Computer Graphics, Computer Networks, Database.

WORKING EXPERIENCES

Max

June 2021 - Current

Engineering Lead, Context Platform

Seattle, WA

- Led a team of 6 engineers to develop a Stateless Context platform, supporting the Max Launch with features like API discovery, Context Hydration, and distributed app-side caching.
- Defined and implemented granular User Context Data with a Graph-based Context Dependency Management system. Achieved a 90% reduction in Redis/DB costs while handling 480K RPS in the U.S.
- Pioneered a server-driven approach for app bootstrap, UX refreshes, and user consent collection, eliminating the need for hardcoded app logic.
- Designed a session-based in-app testing tool named "Multiverse", which streamlined testing for unlaunched markets, content premieres, and fault injection in a live environment without additional setup.

HBO Max

June 2019 - June 2021

Staff Software Engineer, Identity & Auth

Seattle, WA

- Built HBO Max identity platform with microservice architecture, led the U.S. launch, migrated 40M HBO Now user and later expanded to LATAM & EMEA with roaming support.
- Designed a sharded API Gateway to centralize cross-cutting concerns. Introduced an in-house DSL (based on OpenAPI) that reduced API onboarding time from weeks to minutes.
- Developed the HBO Max Privacy Processor to ensure compliance with CCPA and GDPR.

Amazon

Nov 2016 - June 2019

Software Development Engineer, Financial Ledger

Seattle, WA

- Contributed to Amazon's Financial Ledger and Accounting System by developing a multi-tenant configuration storage system and processing engine named "Flare".
- Engineered an in-memory caching/indexing mechanism with interning for Flare, significantly enhancing the configuration query API speed while maintaining minimal memory overhead.
- Designed and implemented a conflict resolution mechanism to handle concurrent edits in Flare.
- Led the design and implementation of an Excel-like UI for configuration editing within Flare.
- Designed and implemented a Schema Definition feature in Flare to describe transaction schemas.

OPENSOURCE PROJECTS

GNU Compiler Collection

May 2015 - August 2015

Libstdc++ Developer

Pittsburgh, PA

- Implemented fundamental TS: Extend `shared_ptr` to support arrays (N3920)
- Implemented fundamental TS: Polymorphic Allocator (N3525)