# Thomas Gebert

693 Thomas S Boyland St. – Brooklyn, NY – United States
☎ 214-451-7333 • ✉ thomas@gebert.app

## Computer skills

**Python**
**SQL**
**Clojure**
**Haskell**
**Erlang/OTP**
Java
ZeroMQ
C
F#
Go
**Kakfa**
JavaScript
**Apache Spark**
Node.JS
Git
Scala
**FFMpeg**
TLA+

## Education

**Western Governor's University** **Salt Lake City, UT**
*B.S. Computer Science*

**University of York** **York, United Kingdom**
*Ph.D. Computer Science, Formal Methods*
Currently Enrolled

## Experience

**CUNY City Tech** **New York, New York**
*Adjunct Lecturer* *August 2022 – June 2023*

- Taught introductory programming to computer engineering students.
  - Introductory and advanced Python and Java.
- Created and graded homework assignments and tests.
- Coded interactive learning modules in JavaScript and Python for students to practice.
- Worked with professors designing cyber-physical systems to help formalize robotics applications.

**Walmart Global Tech** **New York, New York**
*Staff Software Engineer* *August 2021 – August 2022*

- Built NLU pipelines for the Chatbot on walmart.com utilizing the Microsoft Bot Framework.
  - Utilized the Microsoft Bot Framework, F# and C#.
- Coordinated the release of multi-cluster deployment of chatbot.
- Utilized Microsoft Azure, Linux, and Python to build infrastructure.

**Apple Inc.** **New York, New York**
*Senior Software Engineer* *September 2018 – February 2021*

- Designed and built a telemetry and analytics system for finding potential bottlenecks in the cache indexes utilizing Java, Clojure, Kafka, Apache Spark, and Tableau.
- Fixed issues and bugs in the iTunes server backend.
- Built a Kafka-based buffering service to reduce load on indexing and caching for iTunes.
- Maintained and expanded rule engine for iTunes, utilizing Objective-C and C++.
- Deployed Java code onto Amazon Web Services and Linux.

**Walmart Global Tech (Jet.com)** **Hoboken, New Jersey**
*Senior Software Engineer* *July 2016 – August 2018*

○ Wrote Microservices in F#.
○ Utilized the Microsoft Azure stack.
○ Used Apache Kafka to send data between services.
○ Rebuilt the transactional email system to scale to Jet.com size.
○ Taught the F# language during code bootcamps.

**Tone Mobile** **New York, New York**
*Software Engineer* *September 2015 – June 2016*

○ Write and maintain Erlang backends.
○ Create modules for Ejabberd.
○ Integrate chat server with Node.js backend.

**New York University** **New York, New York**
*Research Scientist* *March 2015 – September 2015*

○ Debug Angular.JS frontends.
○ Debug Scala backends.
○ Help write Haskell backends.
○ Programmatically use FFMpeg for video transcoding.

**Sq1** **Dallas, Texas**
*Application Developer* *May 2014 – February 2015*

**Senico, LLC** **Dallas, Texas**
*Software Engineer* *June 2013 – April 2014*

**Propulsion Labs** **Dallas, Texas**
*Software Engineer* *January 2013 – June 2013*

**Amerinational Management Services** **Orlando, Florida**
*Web Developer* *January 2012 – December 2012*

**Lockheed Martin** **Orlando, Florida**
*Software Engineering Intern* *May 2011 – August 2011*

## Public Speaking

**Lambda Days 2023 – Why Design Your Own Levels When Your Computer Can Do it?**: Presented an introduction to doing WebGL with ClojureScript, in addition to showing a few simple procedural level generation algorithms for games

**Lambda Days 2022 – Predicting and Preventing Chaos with Formal Methods in TLA+**: Presented in Krakow, Poland, an introduction to formal methods via a brief description and demonstration of the TLA+ specification language.

**Lambda Days 2020 – Distributed Hash Tables, Video, and Fun!**: Presented in Krakow, Poland, the same talk as stated above.

**Clojure Conj 2019 – Distributed Hash Tables, Video, and Fun!**: Presented in Durham, North Carolina, a demonstration of a project involving a video sharing system using distributed hash tables and to farm out video transcoding.

## Side Projects

**FSharp.Csv**: A reflection-based CSV serializer, written in F#, designed to handle large, multi-gigabyte CSV files, while providing a simple, functional interface, and remaining relatively fast.

**Vertigo.Json**: A reflection-based JSON serializer and deserializer, designed to be used with F#, with an emphasis on being easy-to-use, fast, and null-safe.

For more, please visit gitlab.com/tombert