# Uma **Zalakain**

SOFTWARE ENGINEER · FUNCTIONAL PROGRAMMING

✉ ping@umazalakain.info  |  ⌂ umazalakain.info  |  umazalakain

I bring a unique blend of academic insights and practical experience to any Scala or functional programming role. During my time in academia, I've specialised in advanced functional programming for compiler design. During my time in industry, I've leveraged foundational functional programming principles to drive product development and simplify complex systems.

## Professional Experience

### ITV PLC
*London*

SCALA ENGINEER *July 2022 - present*

Product ITV News serves an average of over 1 million daily page views. Purely functional backend Scala stack using cats, cats-effect and fs2. Daily use of Kafka, PostgreSQL and Redis. Delivery through GA and Jenkins. Deployment through AWS, Docker, Kubernetes and Terraform. RESTful and GraphQL APIs. Significant emphasis on testing, including integration tests and property-based tests. Product development following agile processes. Role Actively led discussions around architectural solutions. Brought forward initiatives that generalise existing ad-hoc solutions into more powerful principled ones. Created tasks aiming to strengthen backend product development. Re-engineered systems and codebases to simplify structure. Directly coordinated with stakeholders and responded to external bug reports. Assisted colleagues with functional programming queries, fostering a collaborative learning environment.

### University of Edinburgh
*Edinburgh*

RESEARCH ASSISTANT *2021*

Product RISE: a functional pattern-based data-parallel language. A high-level functional Scala DSL gets compiled to high performance C, OpenMP, OpenCL and CUDA. The program is rewritten following user-defined optimisation strategies. Role Led refactoring efforts to consolidate the existing codebase and addressed longstanding issues. Interfaced with researchers actively developing new features.

### Microsoft Research Cambridge
*Cambridge*

RESEARCH INTERN *2021*

Product Research on supporting *type changing operations* for *distributed data structures*. Akin to a real-time collaborative text editor, but where the data is structured and typed instead of plain text. The operations executed on the data need to be well typed and enable type changes. Role Placement in the Calc Intelligence group, took part in their day-to-day activities. Modelled a distributed system with type changing operations, defined some of its desirable properties and mechanised proofs that show they hold.

## Academic Qualifications

### PhD Computing Science, unfinished
*Glasgow*

UNIVERSITY OF GLASGOW *2019 - April 2022*

Machine verification of typed process calculi. Modelled typed process calculi with support for linear types, defined type safety properties, and proved they hold. Used theorem provers, dependent types and advanced functional programming techniques to mechanise the models and proofs, and verify them correct.

### MSc Computing Science · with Distinction
*Glasgow*

UNIVERSITY OF GLASGOW *2018 - 2019*

### BSc Hons Computer Science · First class honours
*Glasgow*

UNIVERSITY OF STRATHCLYDE *2014 - 2018*

Awarded Andrew McGettrick prize for outstanding performance throughout the degree (2 recipients).

# Articles and Publications

### Co-Contextual Typing Inference for the Linear $\pi$-Calculus in Agda
**(Extended abstract)** Uma Zalakain, Ornela Dardha

[extended abstract]

[presentation recording]

Extended abstract at Workshop on Type-Driven Development (TyDe) 2021

### $\pi$ with leftovers: a mechanisation in Agda
Uma Zalakain, Ornela Dardha

[published version]

[presentation recording]

In Proceedings of Formal Techniques for Distributed Objects, Components, and Systems (FORTE) 2021

### Type-Checking Session-Typed $\pi$-calculus with Coq
Uma Zalakain, supervised by Ornela Dardha

[thesis]

MSc Thesis, University of Glasgow, 2019

### Evidence-Producing Problem Solvers in Agda
Uma Zalakain, supervised by Conor McBride

[thesis]

BSc Thesis, University of Strathclyde, 2018

# Research Activities

| | | |
|---|---|---|
| 2021 | **PLACES** Program committee member | |
| 2021 | **TyDe** sub-reviewer | |
| 2021 | **PLDI Artifact Evaluation Committee** Program committee member | |
| 2021 | **ICE** Program committee member | |

# Invited Talks

### Theorem Proving with Dependent Types in Agda
Formal Analysis, Theory & Algorithms Seminar, 2021

### An Introduction to Session Types
Mathematically Structured Programming 101 Seminar, 2020

### Mechanising the Linear $\pi$-Calculus
Languages, Systems, and Data Seminar, 2020

### $\pi$ with leftovers: a mechanisation in Agda
- Programming Languages at the University of Glasgow, 2020
- Verification of Session Types, 2020
- Agda Implementors' Meeting XXXII, 2020

### Machine Verification with Agda
Seminar series at the UoG, 2020

### Type-checking session-typed $\pi$-calculus with Coq
- BehAPI – student talks, Leicester, 2019
- SPLV – student talks, Glasgow, 2019

# Tutoring and Co-Supervision

### Co-supervision of MSc theses
- Empirical study of mechanised session types, *Di Cheng*, 2021
- Typed Operations on Distributed Data Structures, *Peng Zhao*, 2021
- Encoding session types into the linear pi-calculus in Agda, *Yuan Gao*, 2021

### Co-supervision of BSc theses
- Encoding session types into the linear pi-calculus in Agda, *Patryk Kaczmarczyk*, 2020
- Abalone in Haskell, *Jing Lee*, 2020

**Tutoring**

- CS1P, FIRST YEAR PROGRAMMING, 2021
- CS1CT, INTRODUCTION TO COMPUTATIONAL THINKING, 2020

# Language Skills

| | |
|---|---|
| **English** | fluent |
| **Basque** | native language |
| **Spanish** | native language |
| **Dutch** | intermediate |
| **Italian** | beginner |

# Skills

| | |
|---|---|
| **Programming Languages** | Agda, Scala, Haskell, Coq, Python, JS, Java, Rust, C, LaTeX |
| **Services** | PostgreSQL, Kafka, Redis |
| **Remote interfaces** | GraphQL, RESTful APIs |
| **Delivery** | Jenkins, Github Actions |
| **Deployment** | Docker, AWS, Kubernetes, Terraform |
| **Sysadmin** | NixOS & GNU/Linux administration, Bourne shells, Git |