# Software-based Reference Protection for Component Isolation

By

Yuval Yarom

September 12, 2014

A Thesis Submitted for the Degree of
Doctor of Philosophy
In the School of Computer Science
University of Adelaide

# Contents

# List of Tables

# List of Figures

# Abstract

Reference protection mechanisms are commonly used to isolate and to provide protection for components that execute within a shared run-time environment. These mechanisms often incur an overhead due to maintaining the isolation or introduce inefficiencies in the communication between the components. Past research operated under the assumption that some performance loss is an acceptable price for the added security that comes with better isolation. This thesis sets out to demonstrate that good isolation does not imply performance loss.

While numerous models for implementing reference protection have been suggested, there is a lack of a unified terminology that allows the comparison of systems from across the domain. This thesis presents a classification framework that captures the trade-offs present in the design of reference protection. It identifies four main models of reference protection: *complete isolation*, where components do not share references to objects; *object sharing*, where components can share data while still maintaining private, unshared data; *partial isolation*, where components have private, unshared data and an exposed interface that allows other component's indirect access to the private data; and *initial isolation*, where components are isolated when created, but the model allows the programmer to share references without restriction.

Applying the classification to systems providing reference protection identifies a gap in the prior research. Partial isolation promises the level of security expected from component isolation combined with efficient communication. Yet, the only implementation of partial isolation of components uses expensive run-time checks to enforce the protection.

To bridge this gap, this thesis presents the Exported Types design. Exported Types is a type system design that enforces partial isolation at compile time. Using compile-time checks removes the run-time overhead of enforcing the protection model. The design is applied to a meta-circular Java virtual machine to isolate the virtual machine code from the application. Applying reference protection in this scenario reduces the number of classes the virtual machine exposes to the application by two orders of magnitude. Performance tests demonstrate that reference protection, and the higher security it provides, are achieved at no performance cost.

# Declaration

I certify that this work contains no material which has been accepted for the award of any other degree or diploma in any university or other tertiary institution and, to the best of my knowledge and belief, contains no material previously published or written by another person, except where due reference has been made in the text. In addition, I certify that no part of this work will, in the future, be used in a submission for any other degree or diploma in any university or other tertiary institution without the prior approval of the University of Adelaide and where applicable, any partner institution responsible for the joint-award of this degree.

I give consent to this copy of my thesis, when deposited in the University Library, being made available for loan and photocopying, subject to the provisions of the Copyright Act 1968.

I also give permission for the digital version of my thesis to be made available on the web, via the University's digital research repository, the Library catalogue and also through web search engines, unless permission has been granted by the University to restrict access for a period of time.

Yuval Yarom
September 12, 2014

# Acknowledgements


> Ithaka gave you the marvelous journey.
> Without her you would not have set out.
> She has nothing left to give you now.
> —*Constantine Cavafy, "Ithaka"*

The journey described in the following pages would not have been possible without the dedicated guidance of my supervisors Dave Munro and Katrina Falkner. It is through the wisdom they shared, the encouragement they provided and the patience they exercised that this work has come to fruition. It was an honour working with them and the lessons they have taught will stay with me for the rest of my life.

I would also like to thank Henry Detmold who for a long time acted as an unofficial supervisor. Many thanks also to Steve Blackburn, Ron Morrison and John Zigman for the help they provided and their useful suggestions in various stages of this work. Thanks are also due to Richard Jones and to the anonymous examiner for the careful reading of this thesis and for the wise and helpful comments provided.

I am indebted to Kathy Cooper, Cathie Liersch, Sharyn Liersch, Julie Mayo, Jo Rogers and Tracey Young who helped navigating the treacherous waters of the university administration and for providing some geek-free refuge. Thanks also to William Brodie-Tyrrell, Diana Hill, Peter Kelly, Ekim Kocadag, Joseph Kuehn, Matt Lowry, Peter Nguyen, Travis Olds and Stani Ovtcharova who, over the years, shared the working space with me and helped make the time spent on this PhD an enjoyable experience.

The sages teach that without food there is no learning.[1] It would be futile to try listing all fine establishments providing sustenance around the university. However, special mention must be made of the spicy chicken at Raah Cafe, the mayonnaise chicken at Zen Garden and everything at Vego and Love'n It.

Most of all, I would like to thank my wife Karen Gekker and my children Yarden and Itay for their help, support and encouragement throughout the years. I acknowledge and deeply appreciate the sacrifices they made to allow me to continue my studies.


[1] Ethics of the Fathers 3:17 אם אין קמח, אין תורה. Lit.:*If there is no flour, there is no (study of the) Torah.*