Recovering Debug Information from Randomized Code Movement

Steven Neisius, UC Irvine

SoCal PLS, Fall 2013

Motivation

* Many projects have applied diversity to binaries

* A. Homescu, S. Neisius, P. Larsen, S. Brunthaler, and M. Franz; "Profile-guided Automated Software Diversity,"' in 2013 International Symposium on Code Generation and Optimization (CGO 2013), Shenzhen, China; February 2013.

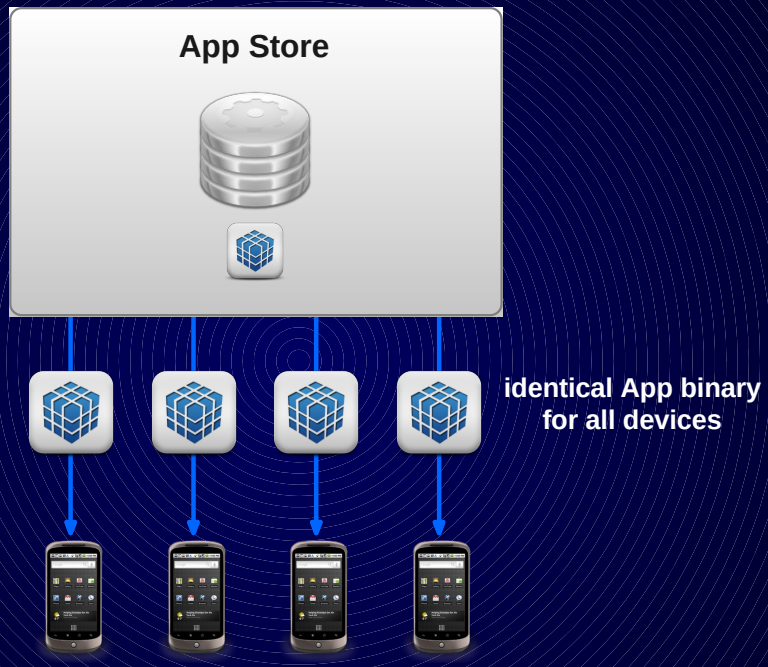
* Richard Wartell, Vishwath Mohan, Kevin W. Hamlen, and Zhiqiang Lin. 2012. Binary stirring: self-randomizing instruction addresses of legacy x86 binary code. In Proceedings of the 2012 ACM conference on Computer and communications security (CCS '12). ACM, New York, NY, USA.

* Etc...

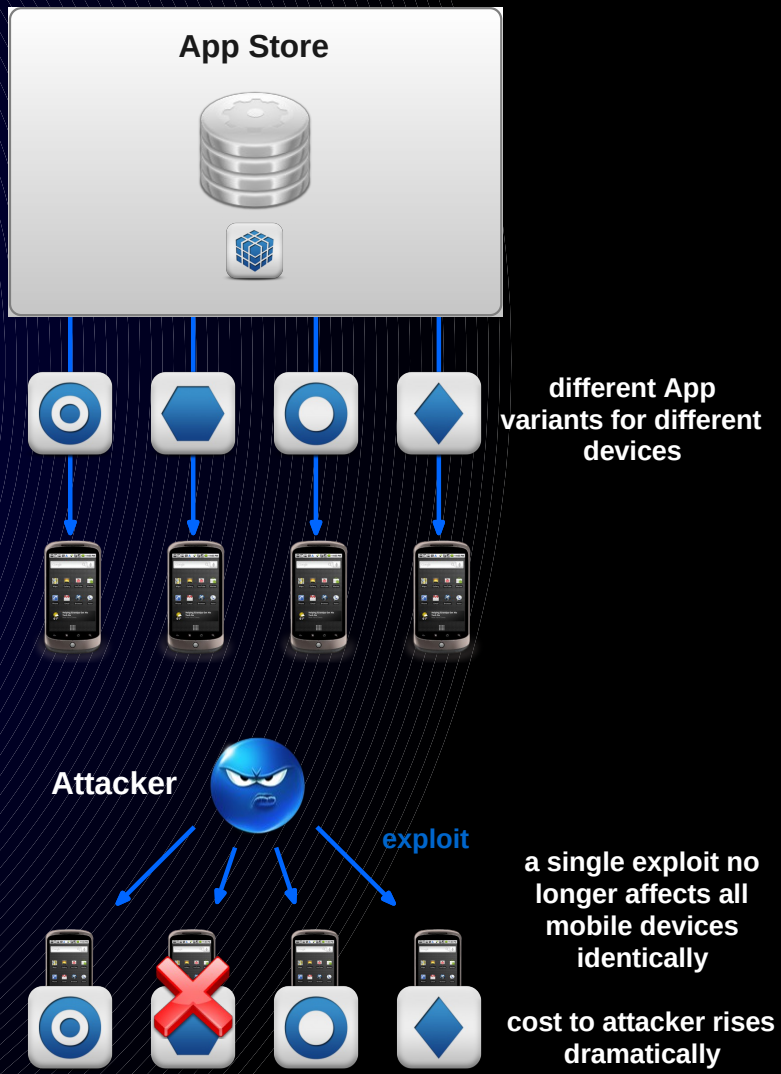
* None have explored crash reporting

Diversity In A Nutshell

Current Practice



With Software Diversity

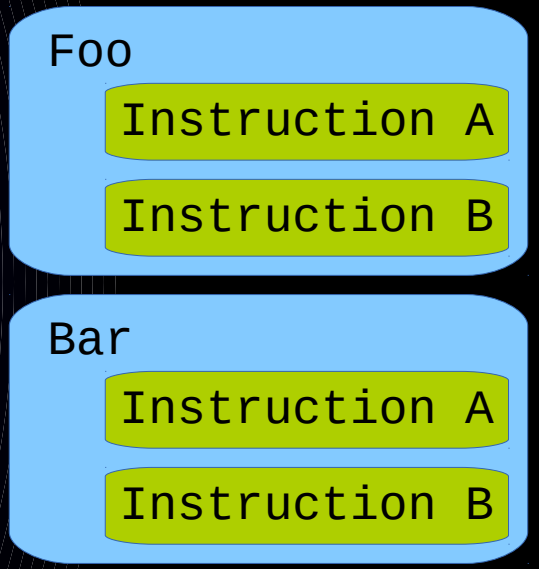


Code Movement Transformations

* NOP Insertion

* Schedule Randomization

* Function Layout

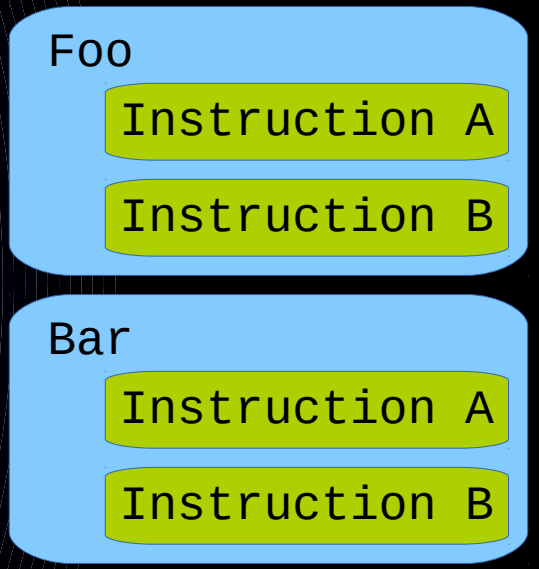


Code Movement Transformations

* NOP Insertion

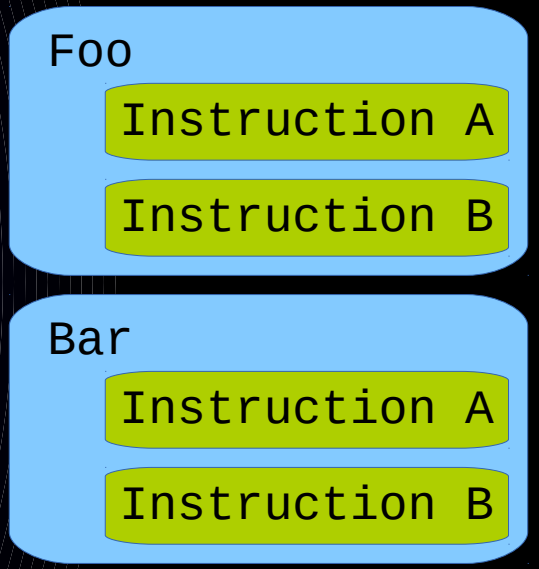
* Schedule Randomization

* Function Layout



Code Movement Transformations

* NOP Insertion



* Schedule Randomization

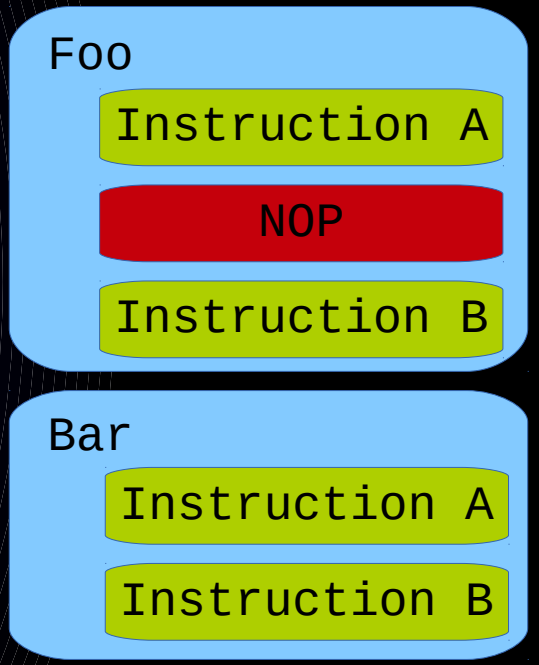
* Function Layout

Code Movement Transformations

* NOP Insertion

* Schedule Randomization

* Function Layout

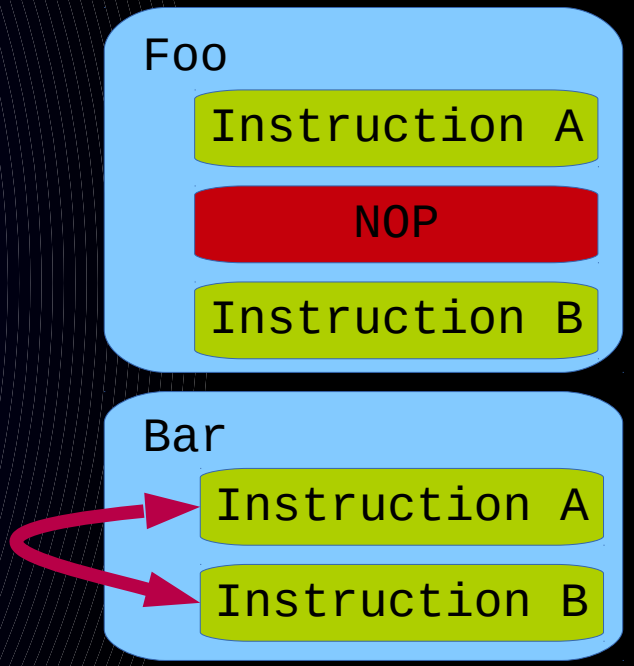


Code Movement Transformations

* NOP Insertion

* Schedule Randomization

* Function Layout

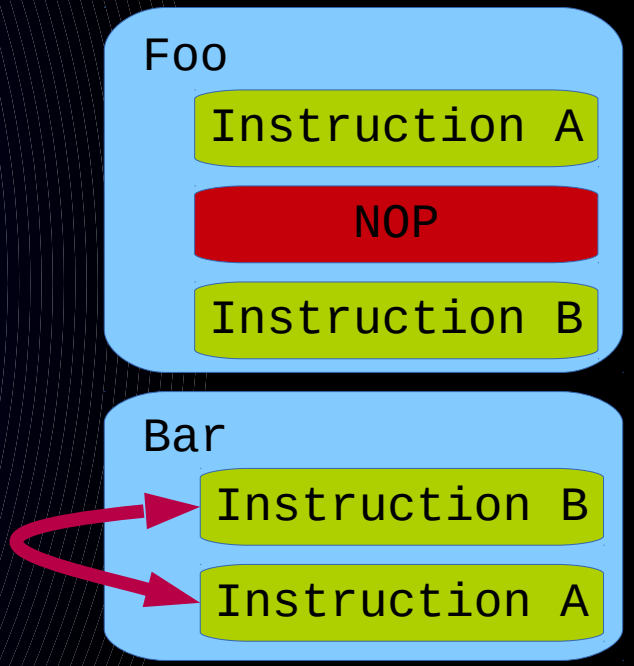


Code Movement Transformations

* NOP Insertion

* Schedule Randomization

* Function Layout

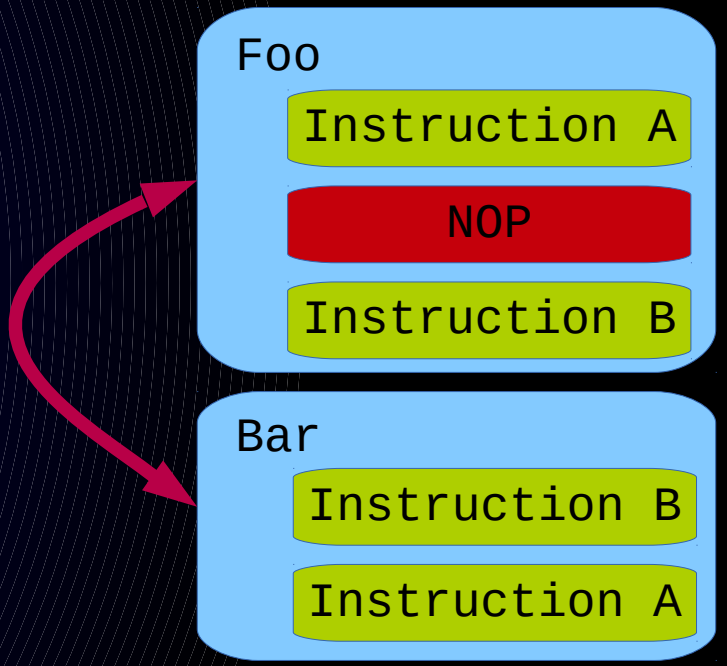


Code Movement Transformations

* NOP Insertion

* Schedule Randomization

* Function Layout

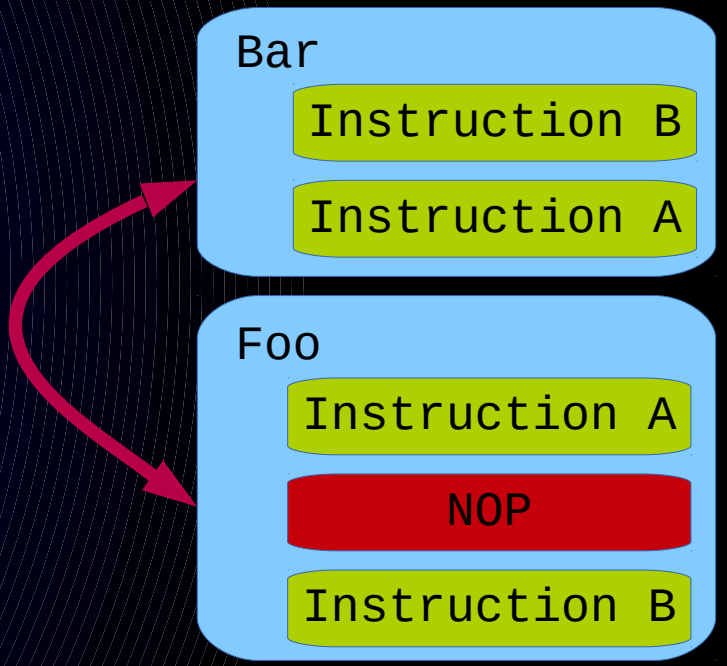


Code Movement Transformations

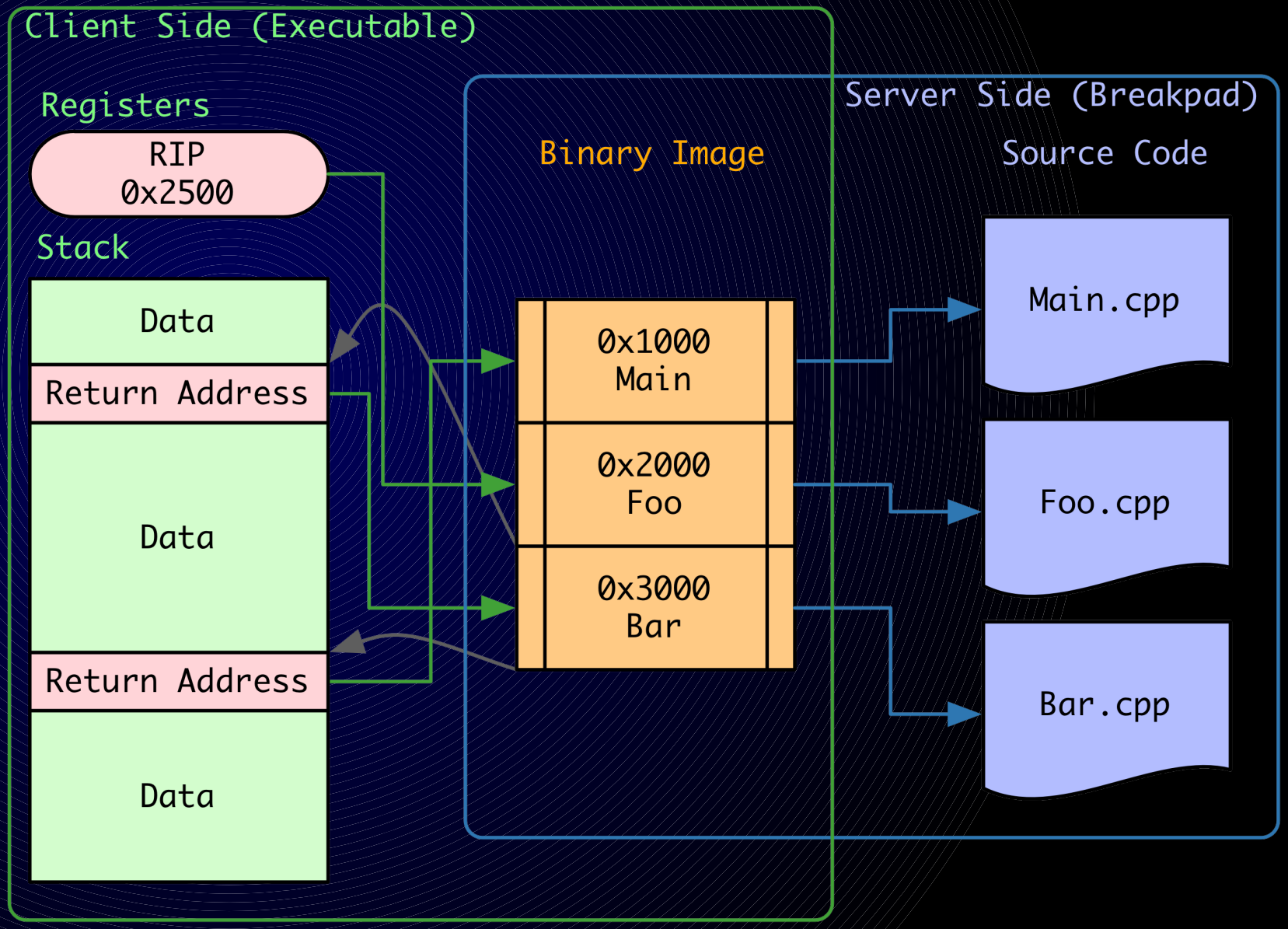
* NOP Insertion

* Schedule Randomization

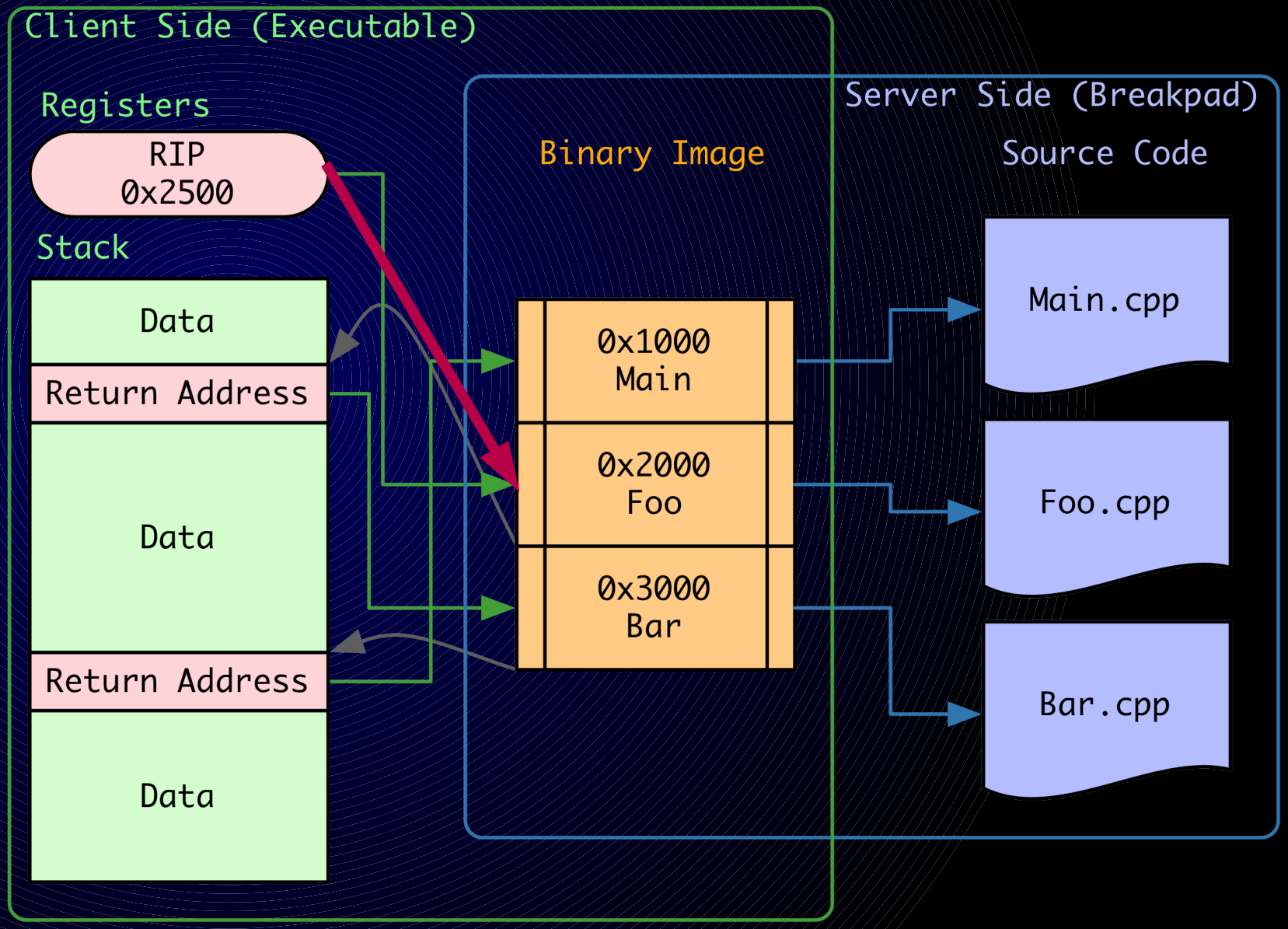
* Function Layout



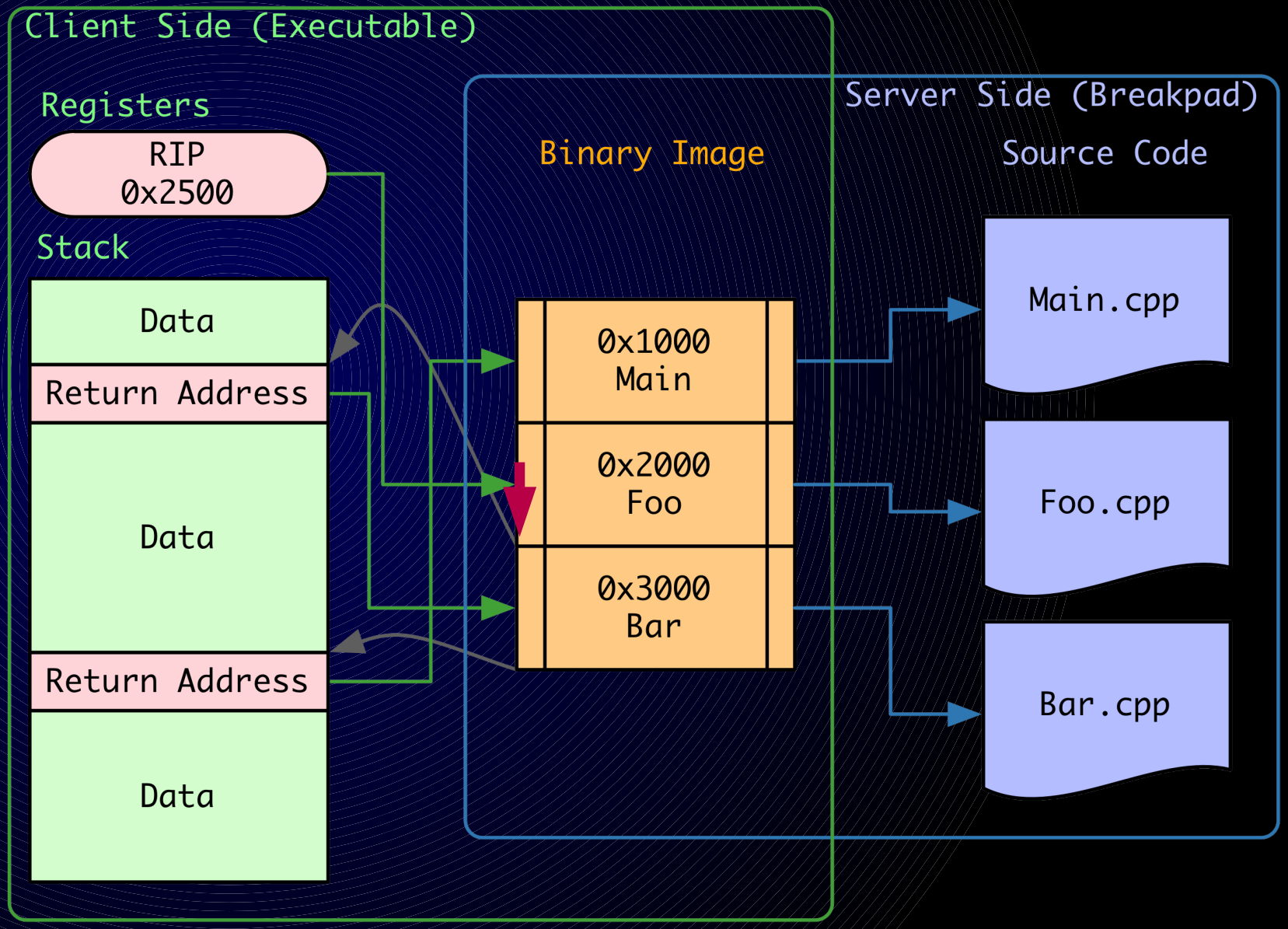
Crash Reporting



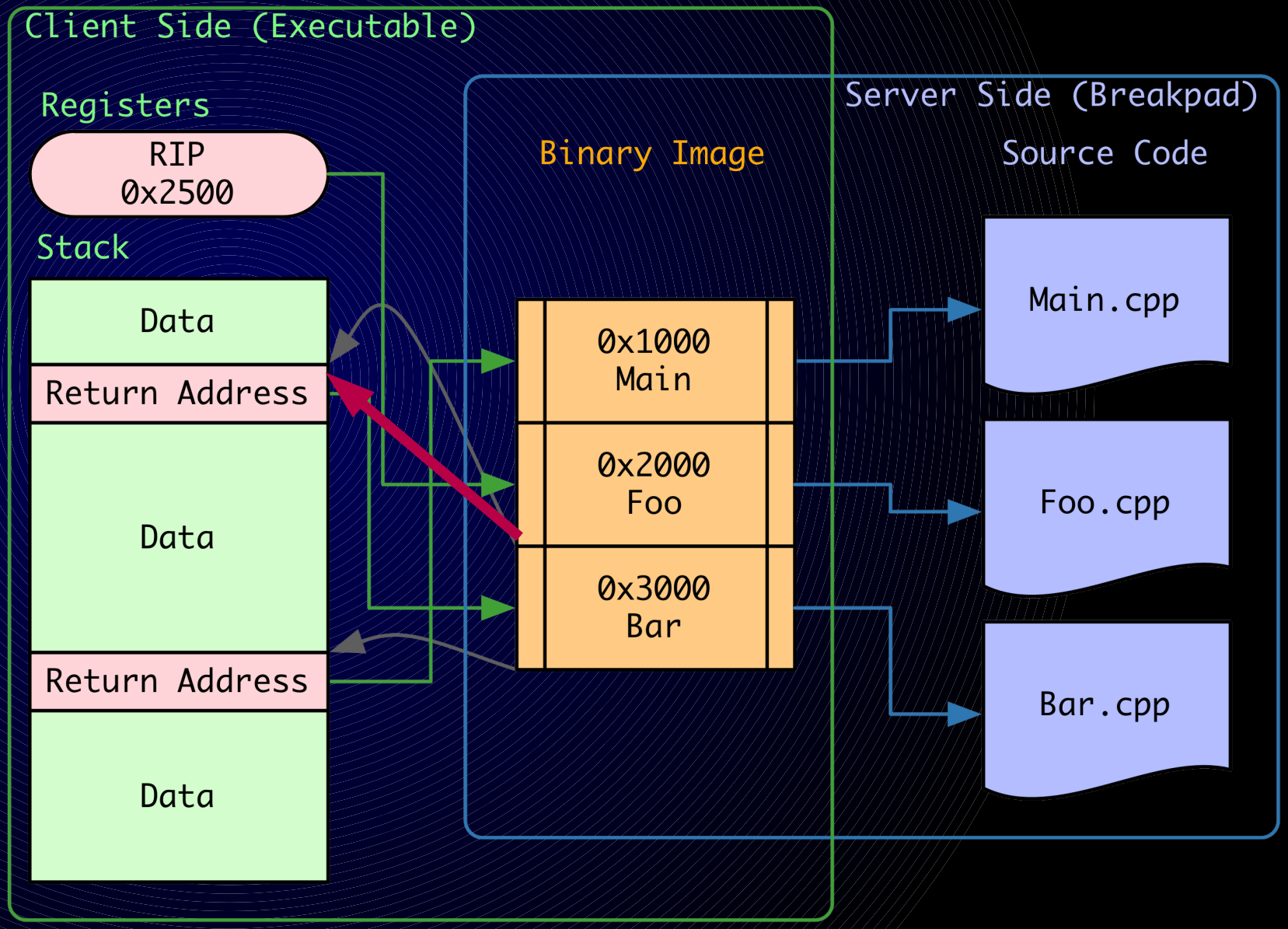
Crash Reporting



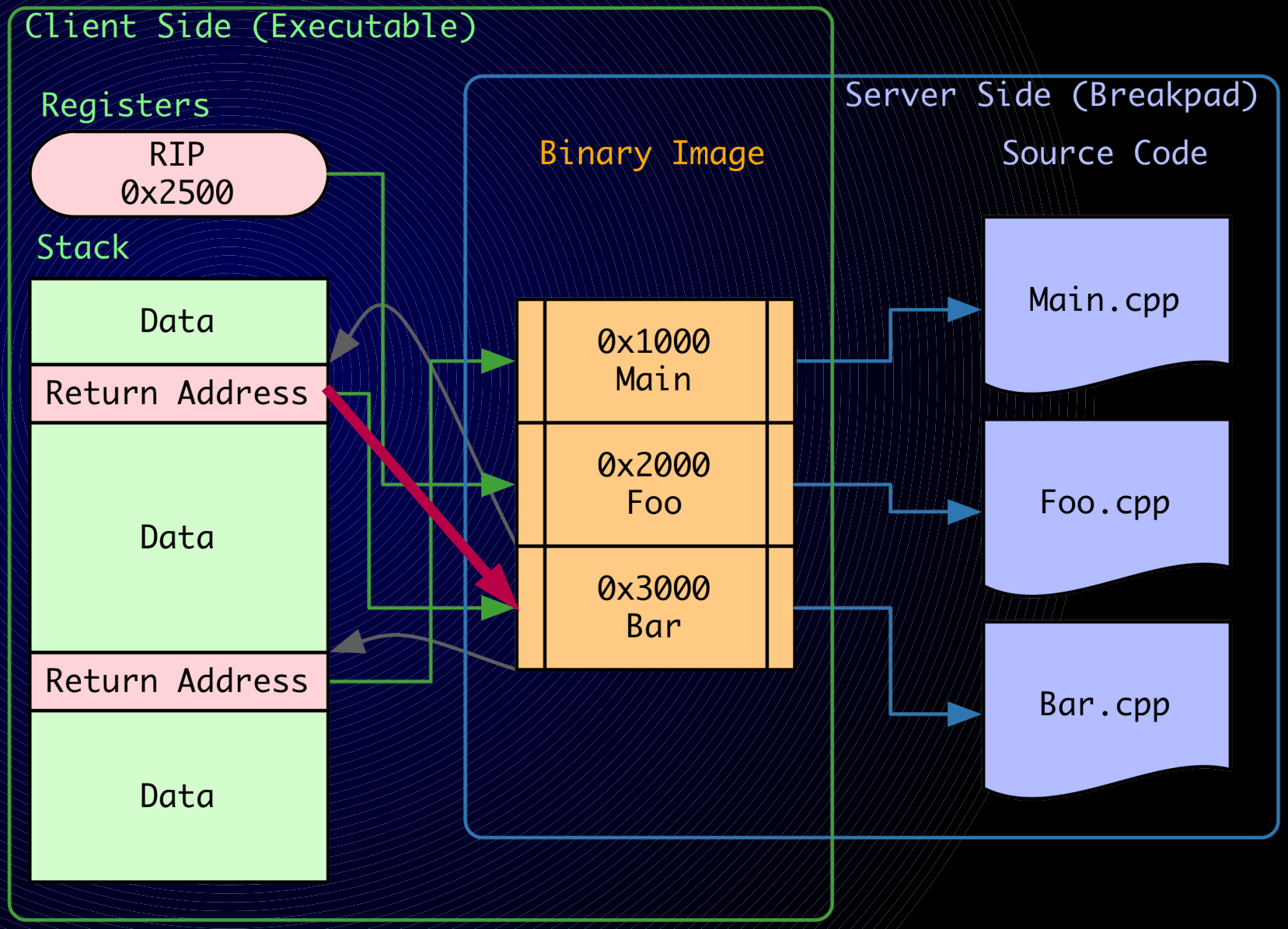
Crash Reporting



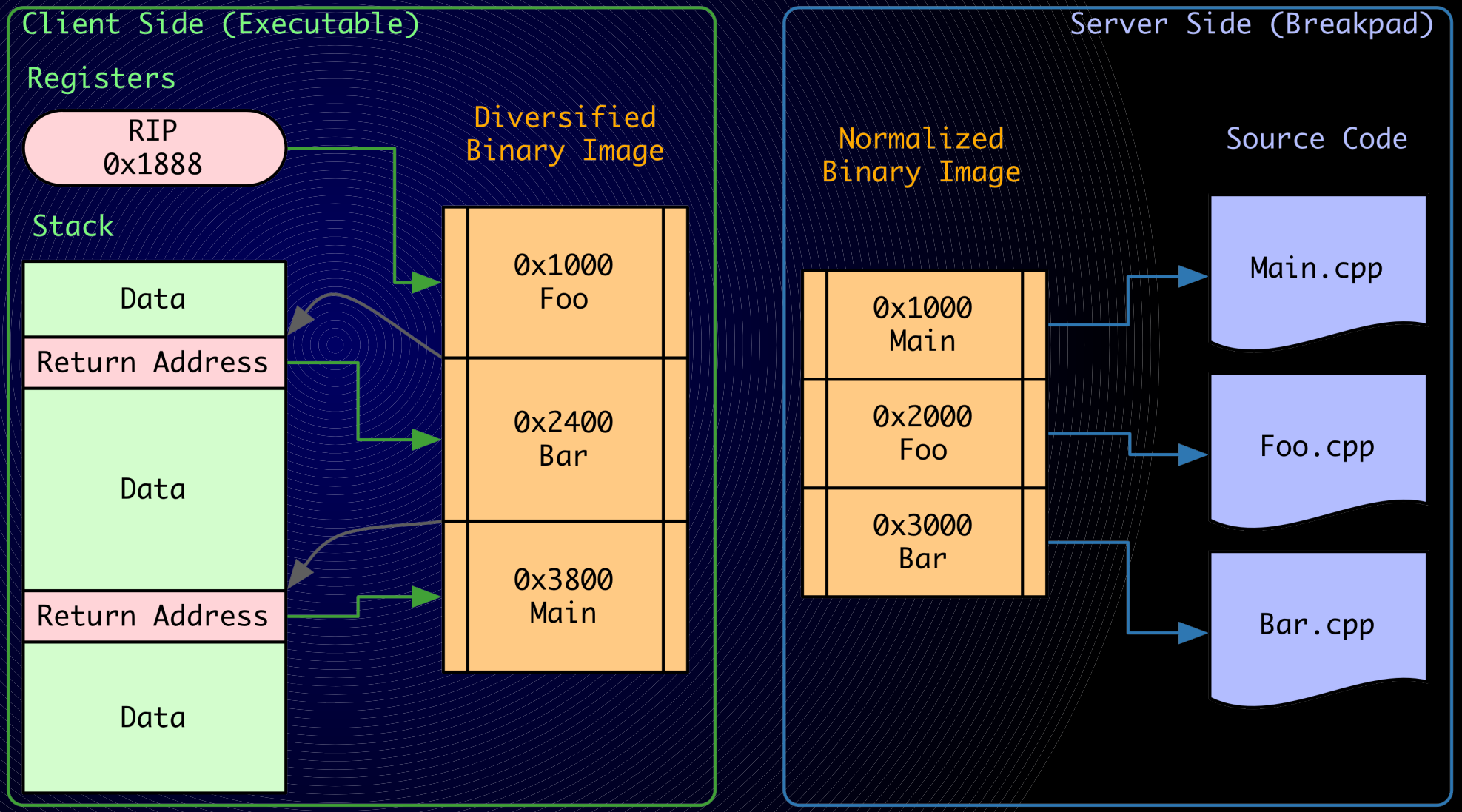
Crash Reporting



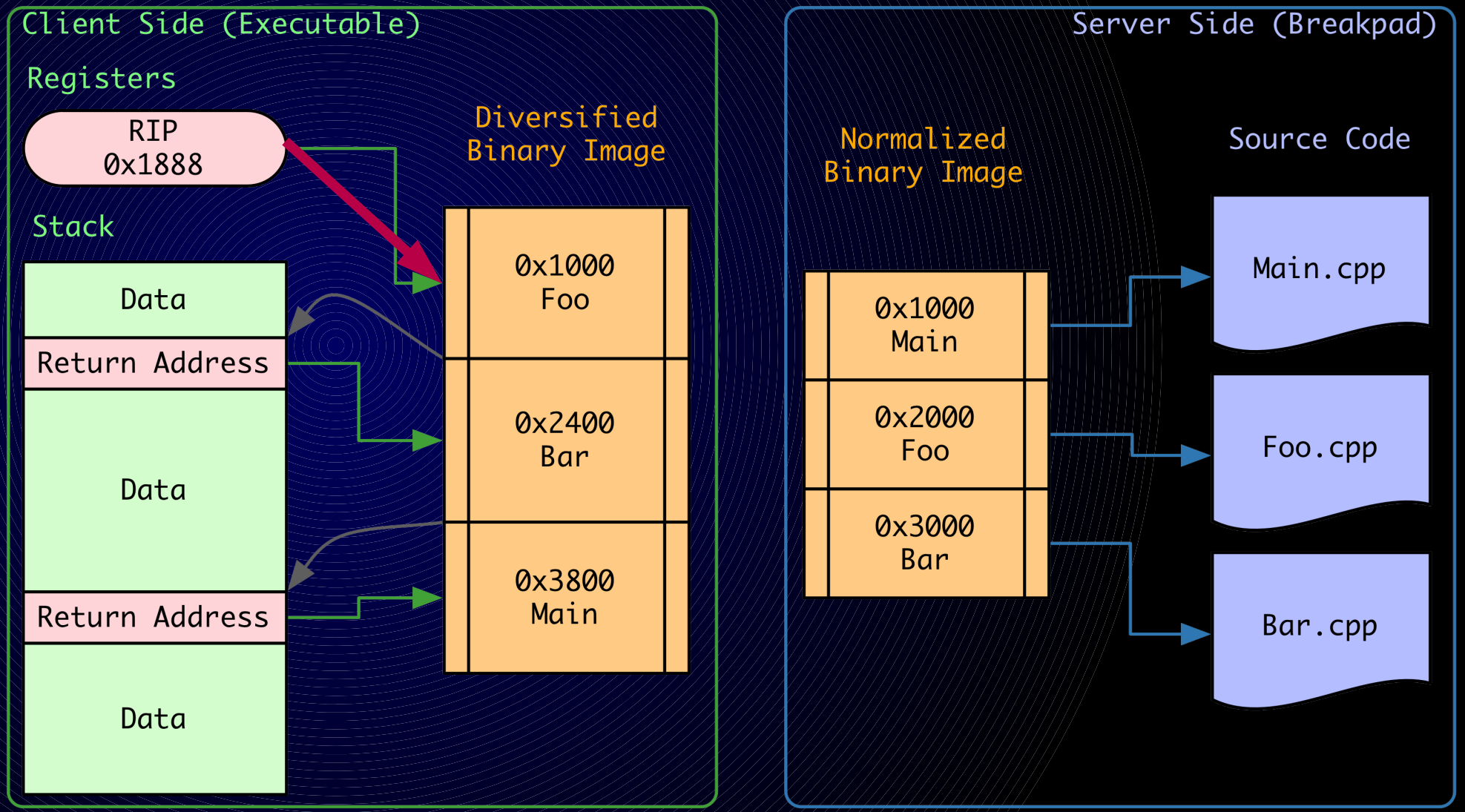
Crash Reporting



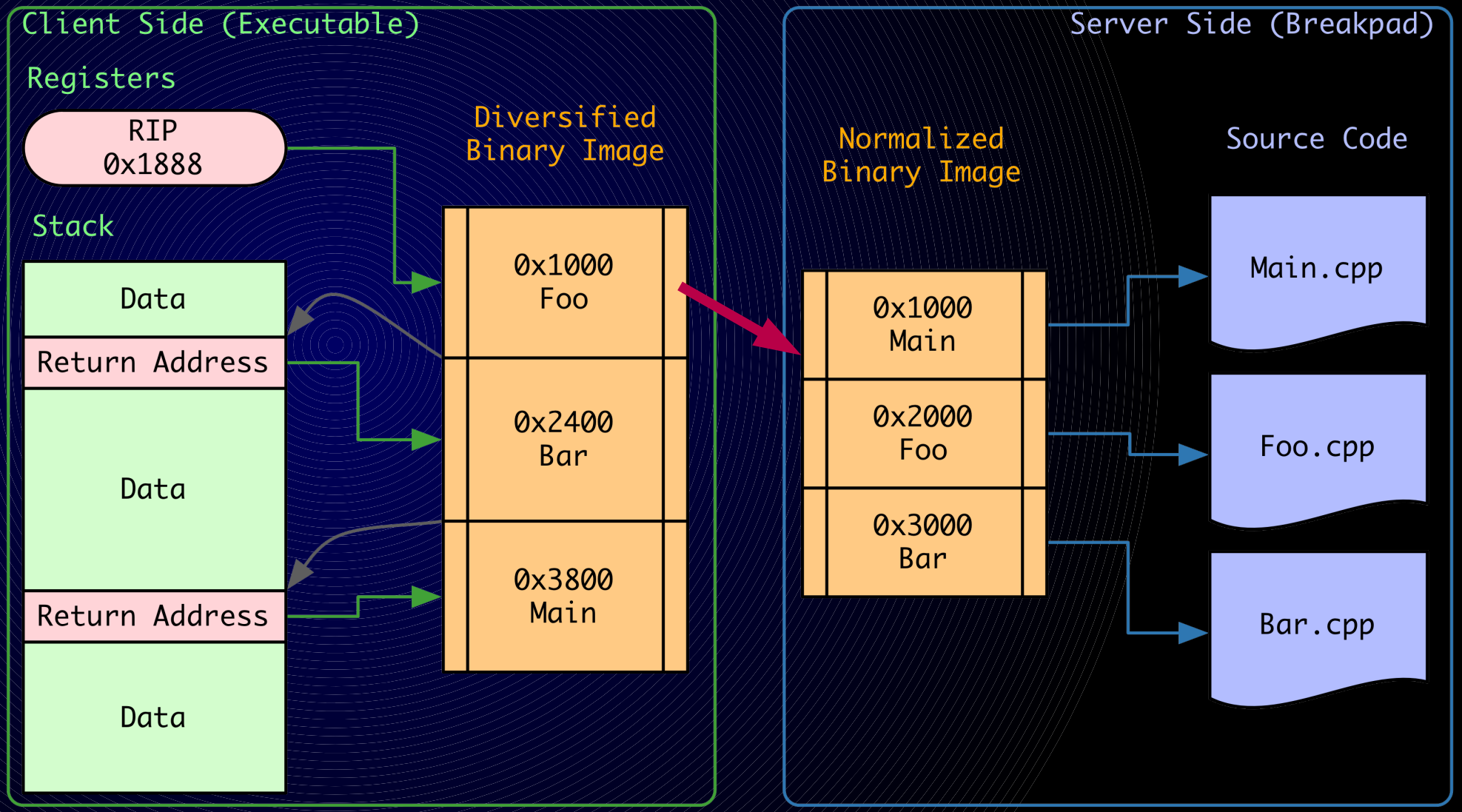
With Diversity



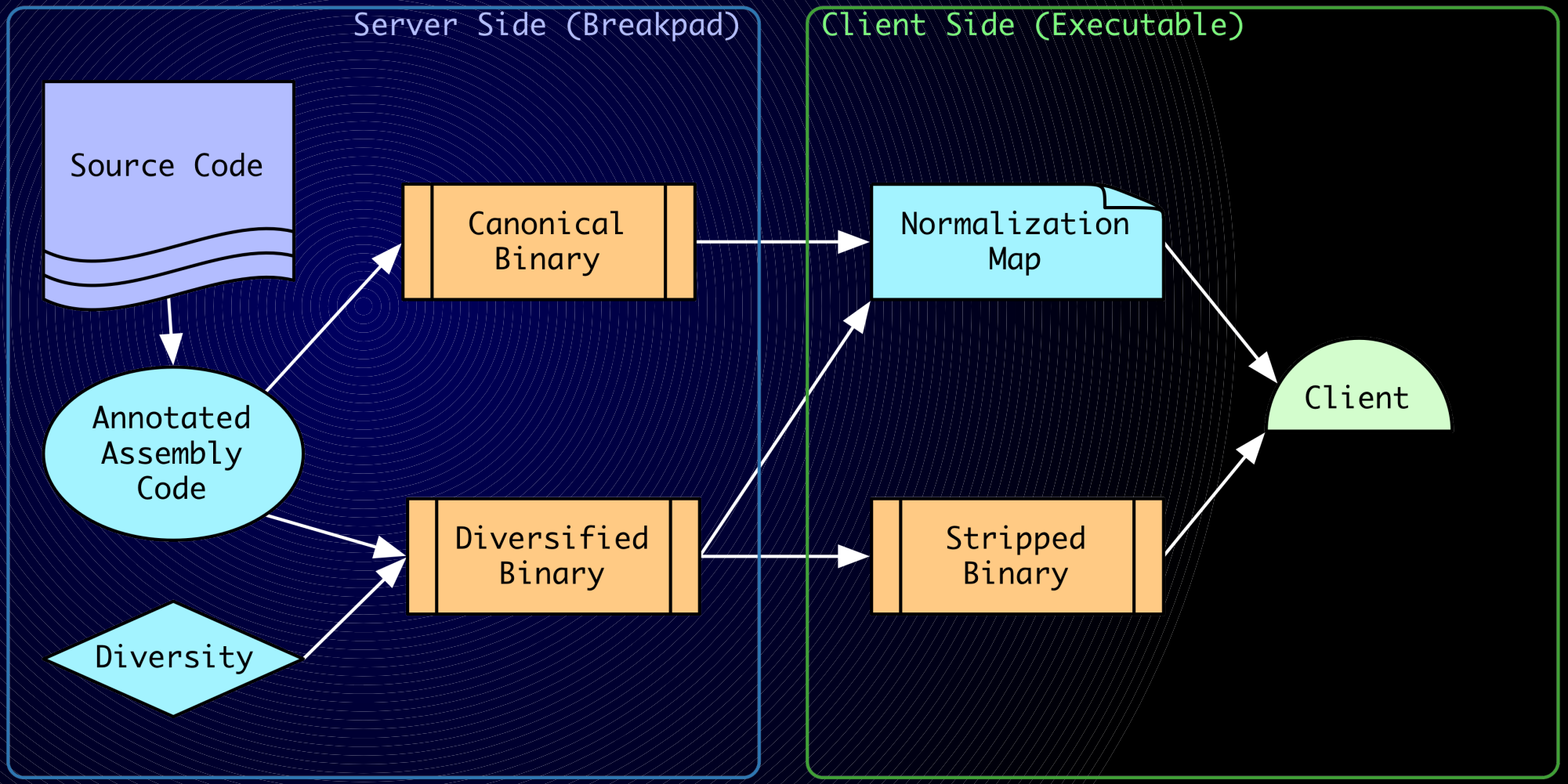
With Diversity



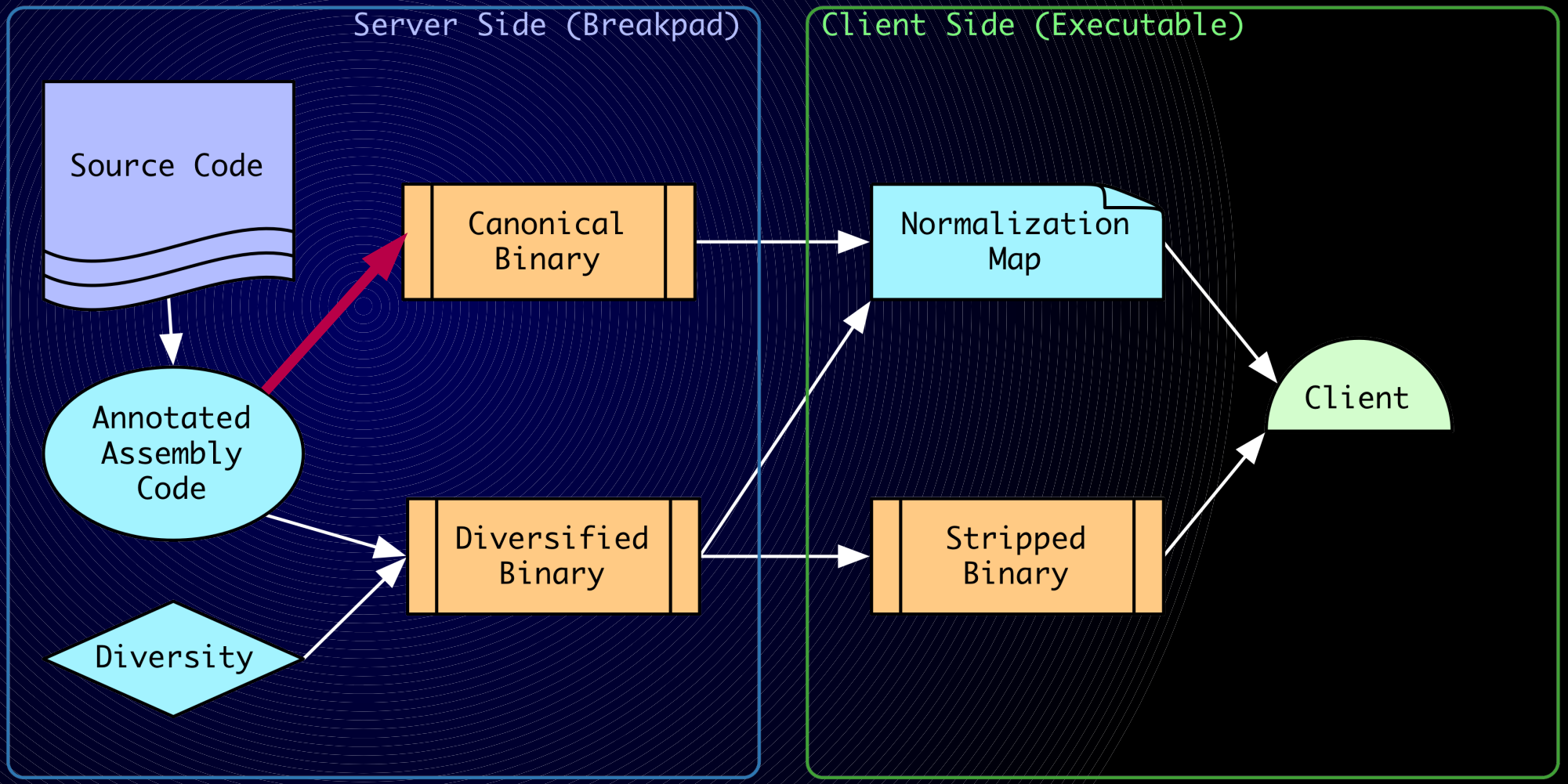
With Diversity



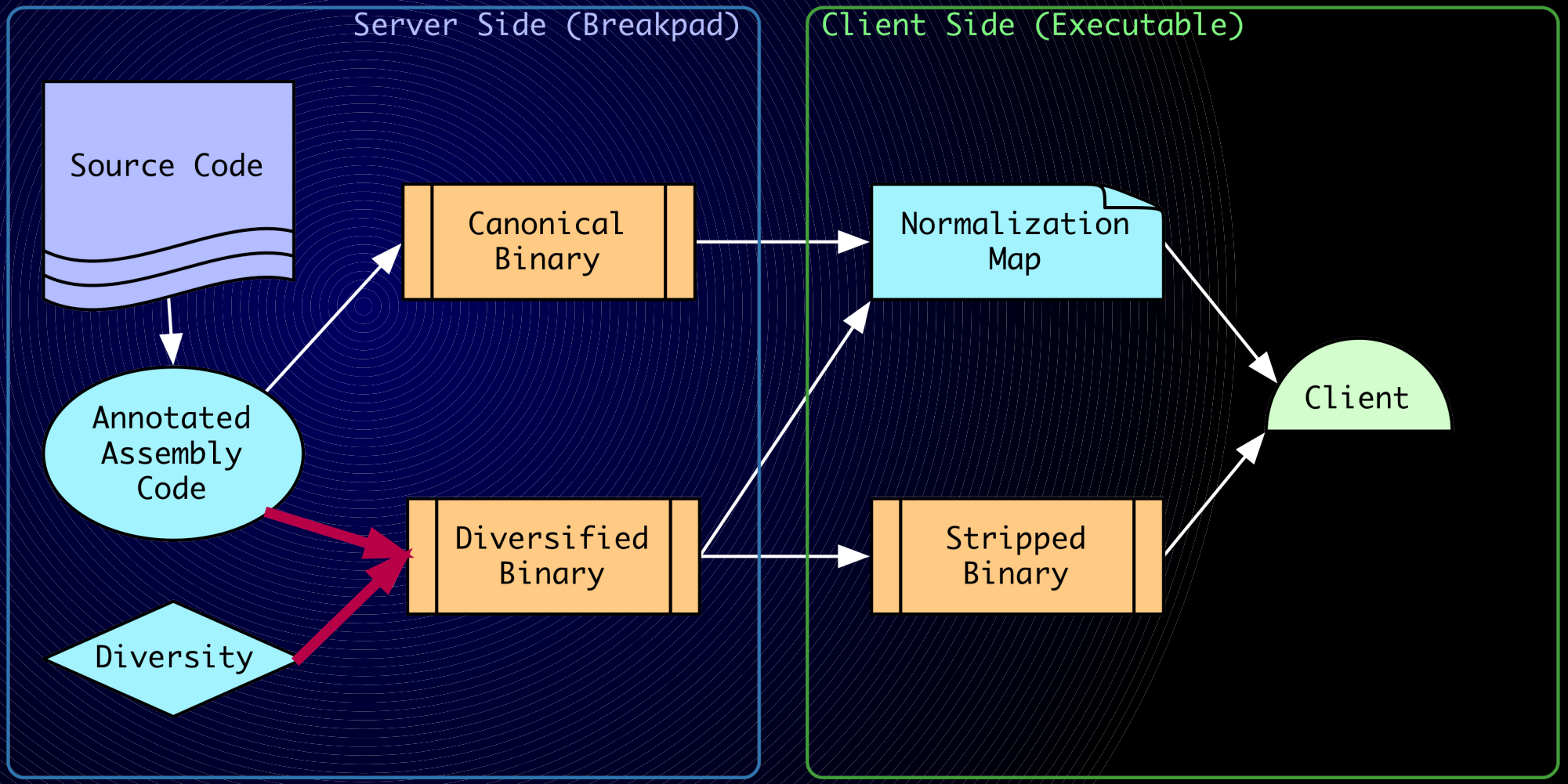
Our Approach



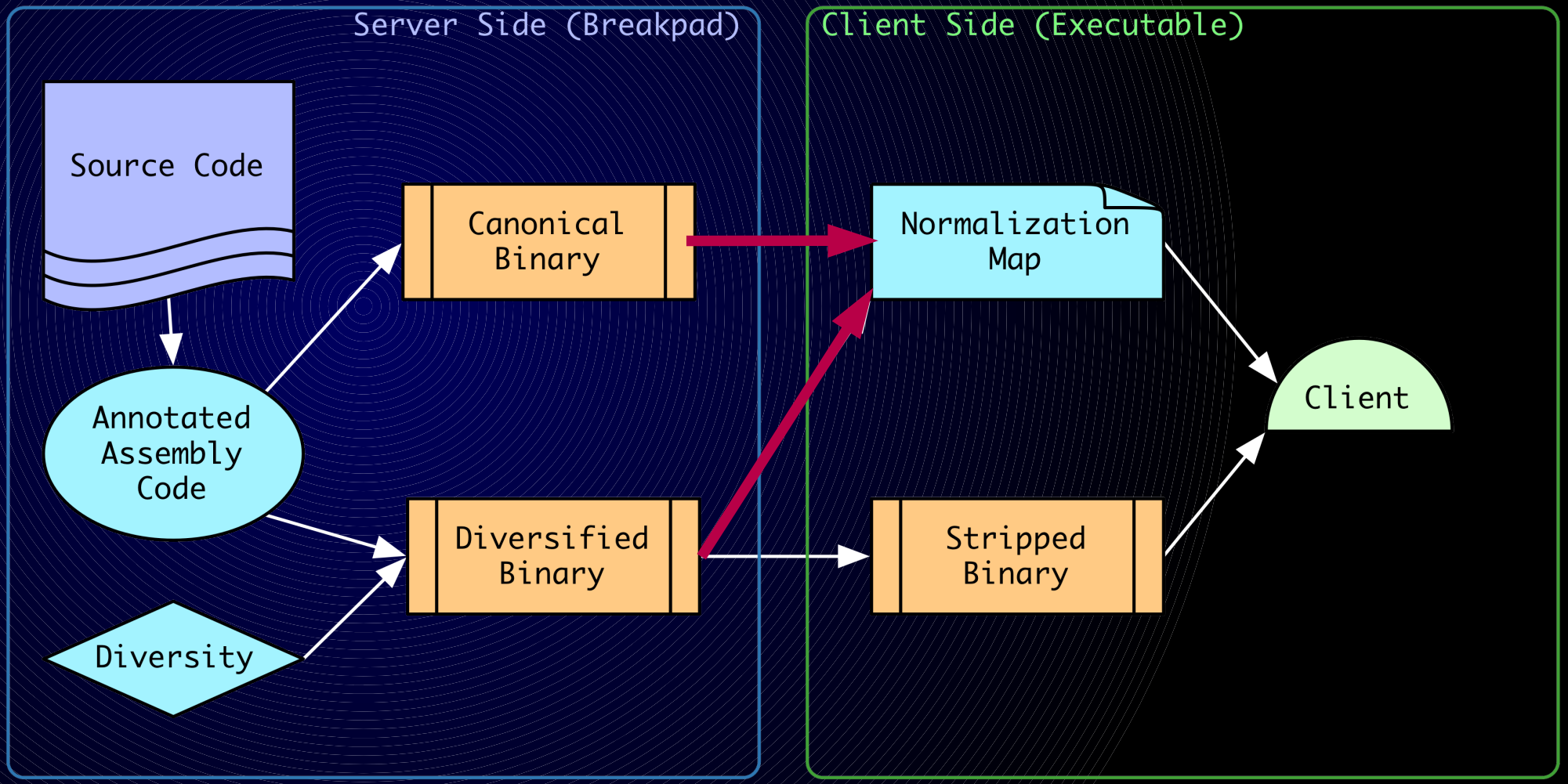
Our Approach



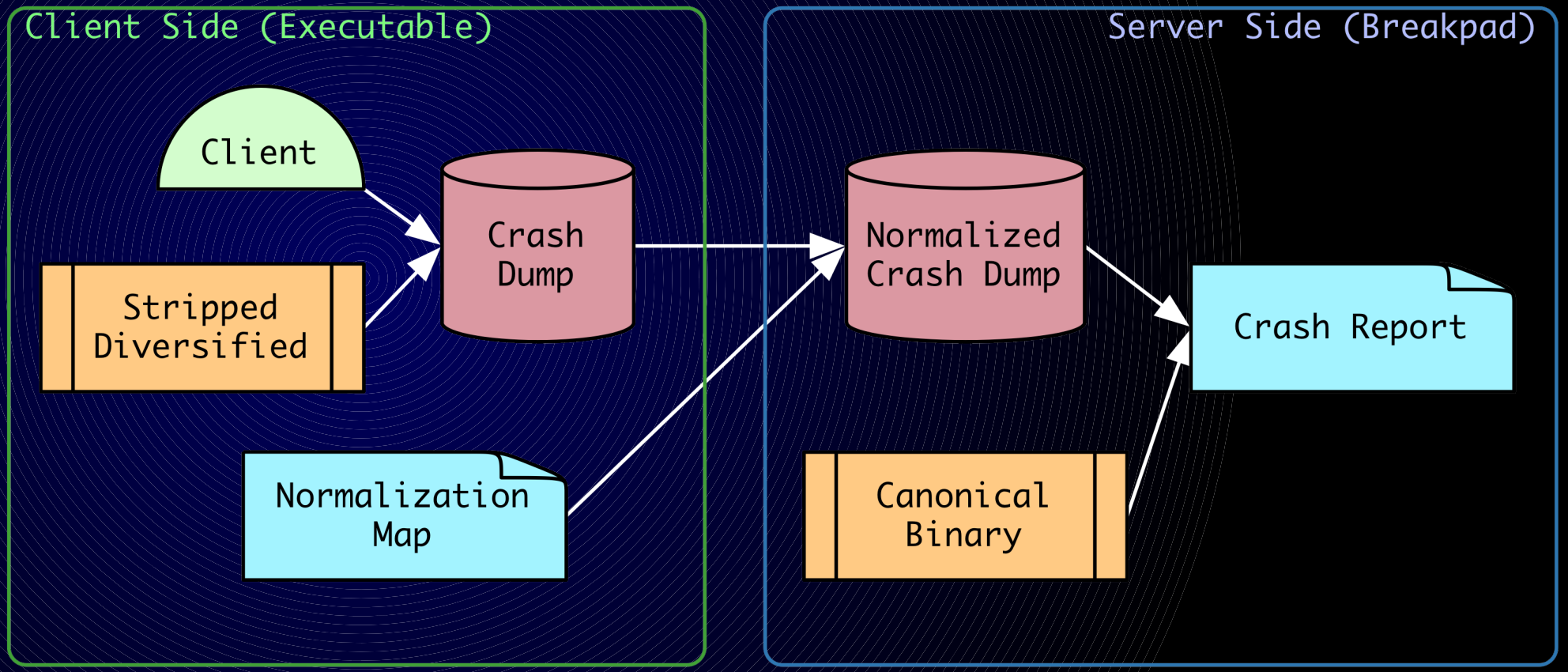
Our Approach



Our Approach



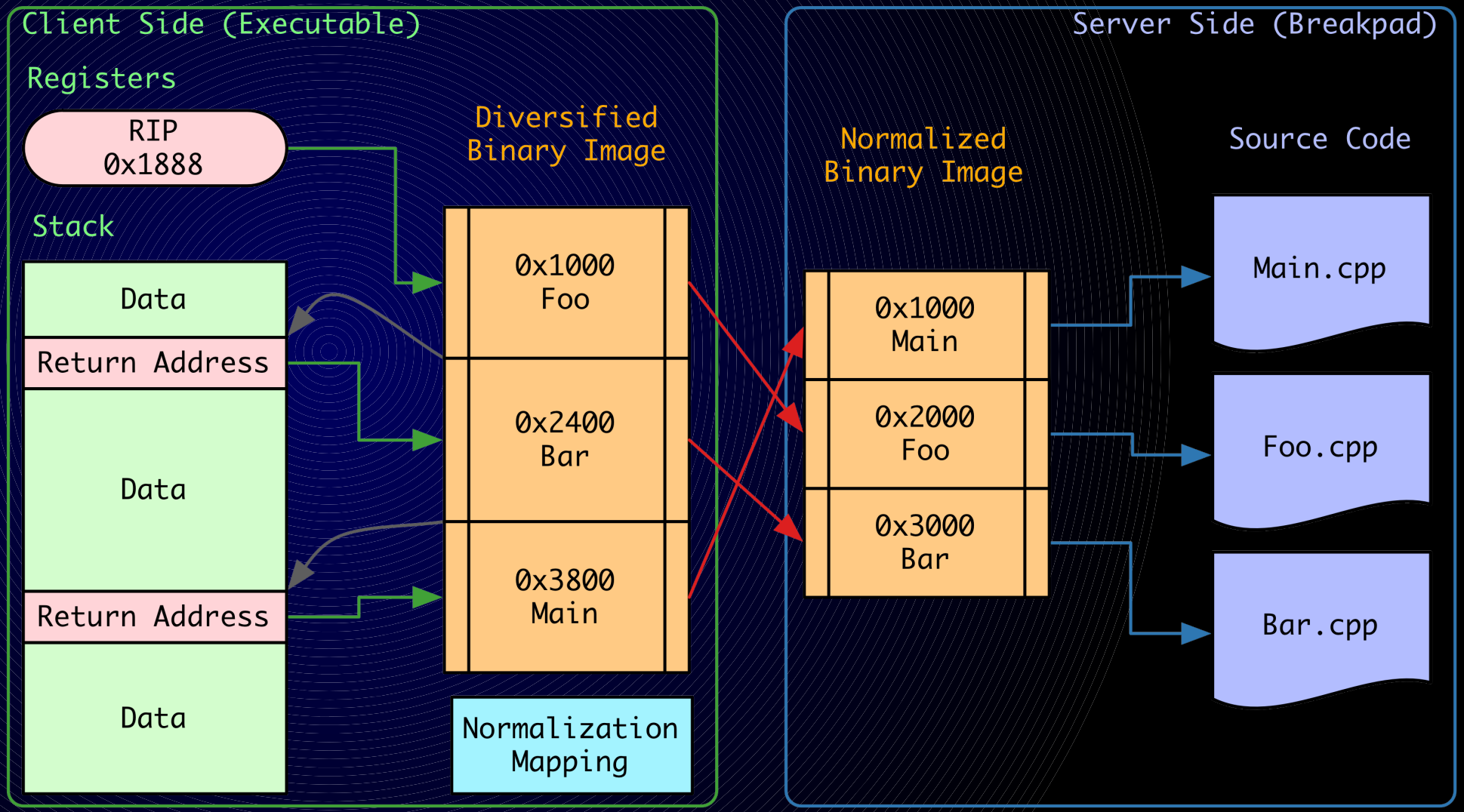
Diversified Crash Reporting



Normalization Locale

Client	Server
+ Privacy	+ Frame info
+ Transparency	- More bandwidth
- More work	- Modify tracker

Normalization Method



Results

* Successfully stackwalk a diversified crash dump as if it was created by the canonical version

* Supports any code movement transformation that operates on assembly code

Future Work

* Optimize mapping time and size

* Mapping for register and stack randomization

* Patching and updating

Questions?

Thanks!