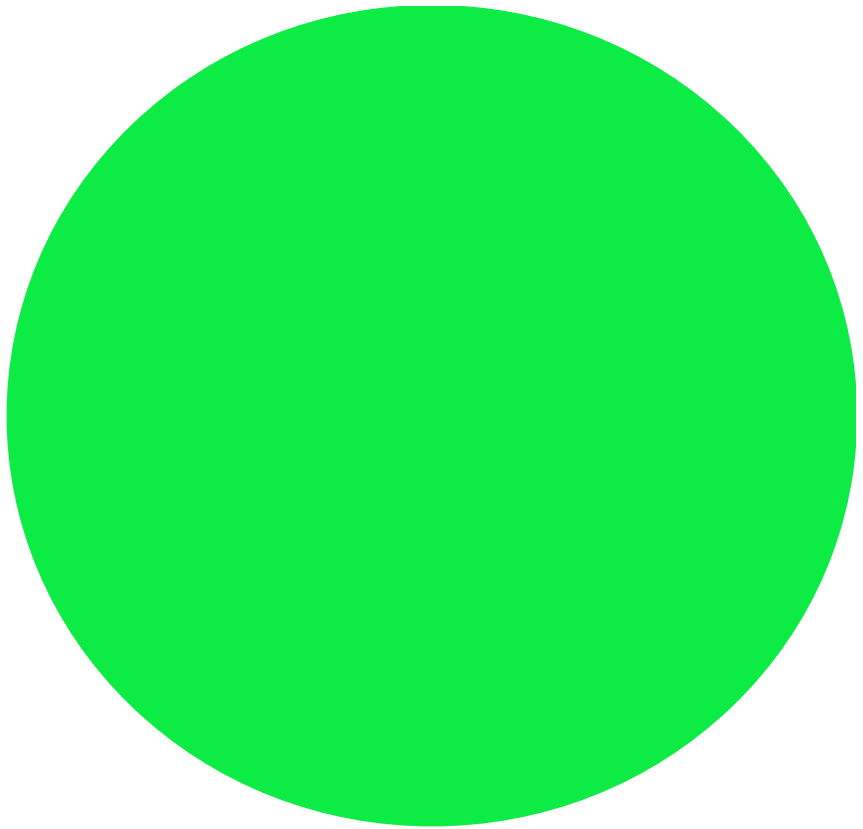
Tailored Source Code Transformations to Synthesize Computationally Diverse Program Variants

Benoit Baudry, Simon Allier, Martin Monperrus

- This talk is about the generation of very large quantities of sosie programs

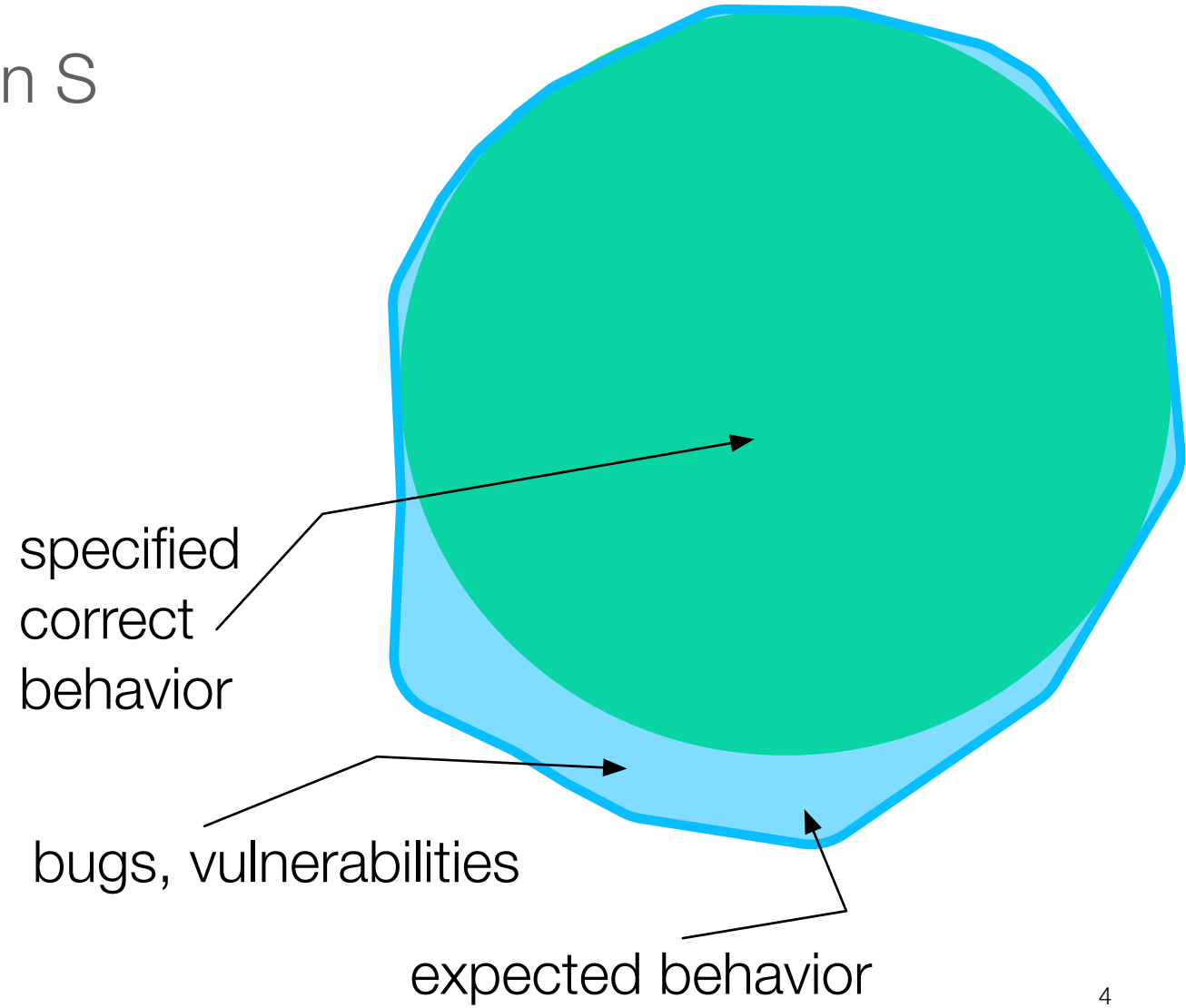
sosie program

- Given a specification S



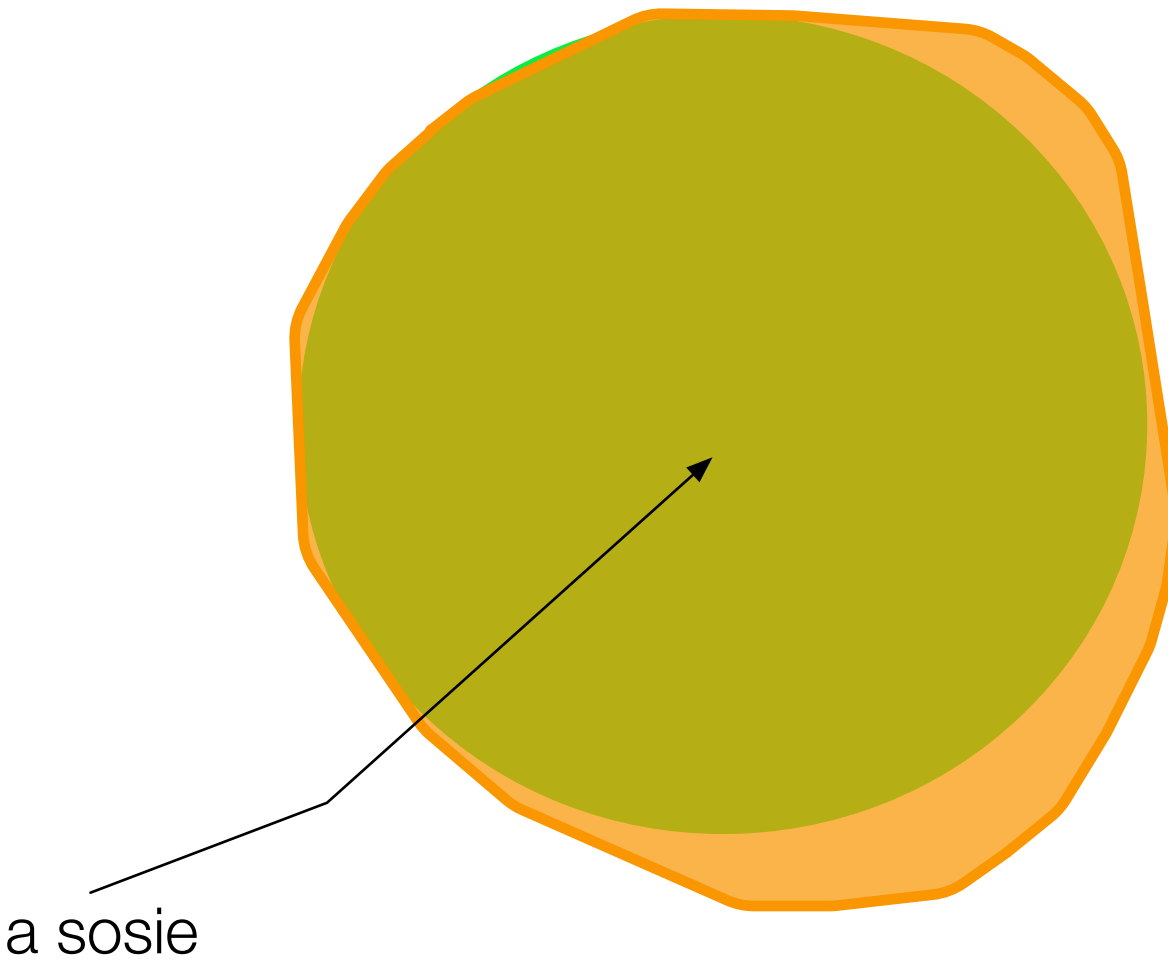
sosie program

- Given a specification S
- Given a program P that conforms to S



sosie program

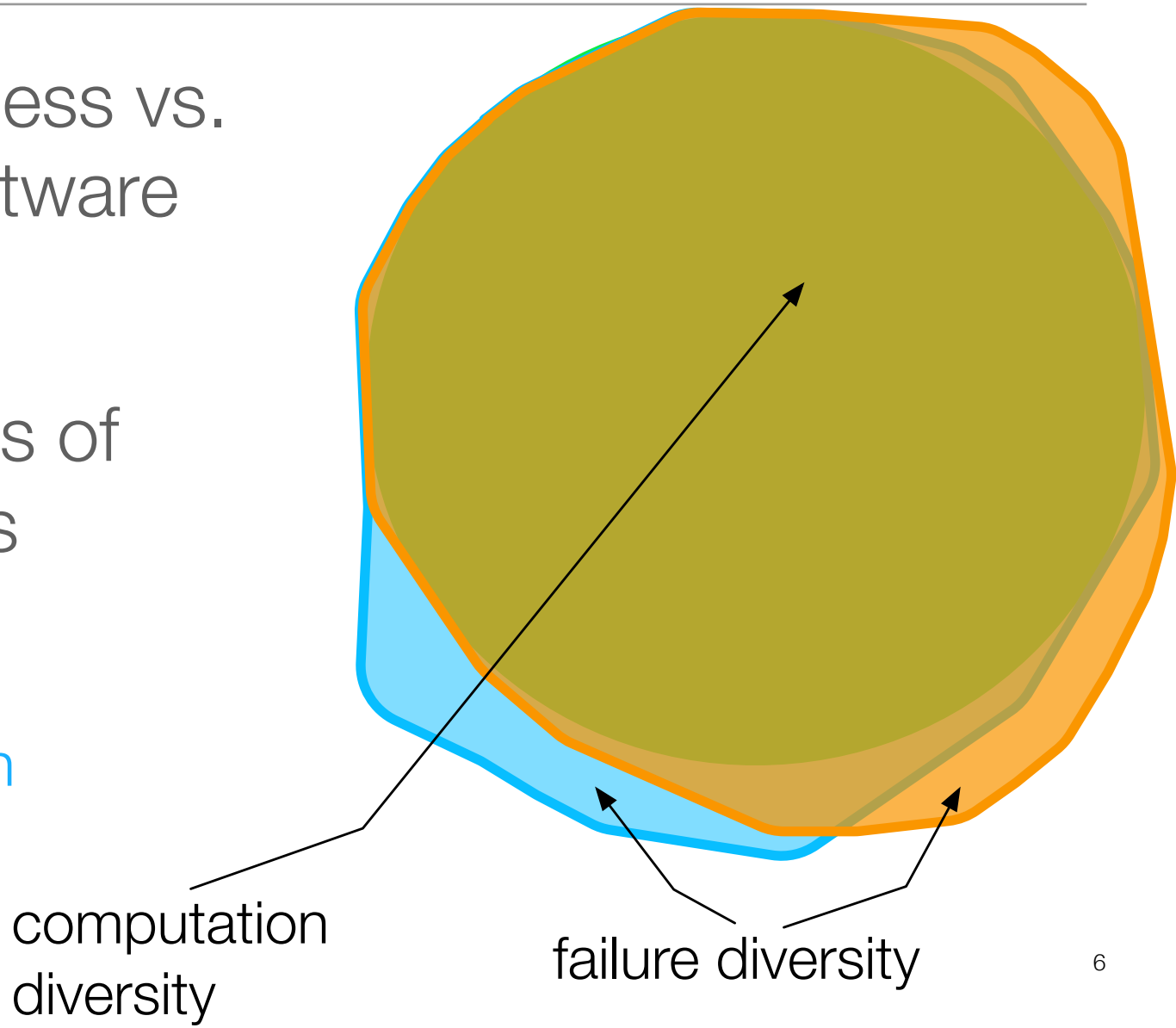
- Given a specification S
- Given a program P that conforms to S
- A sosie of P is a variant of P that also conforms to S



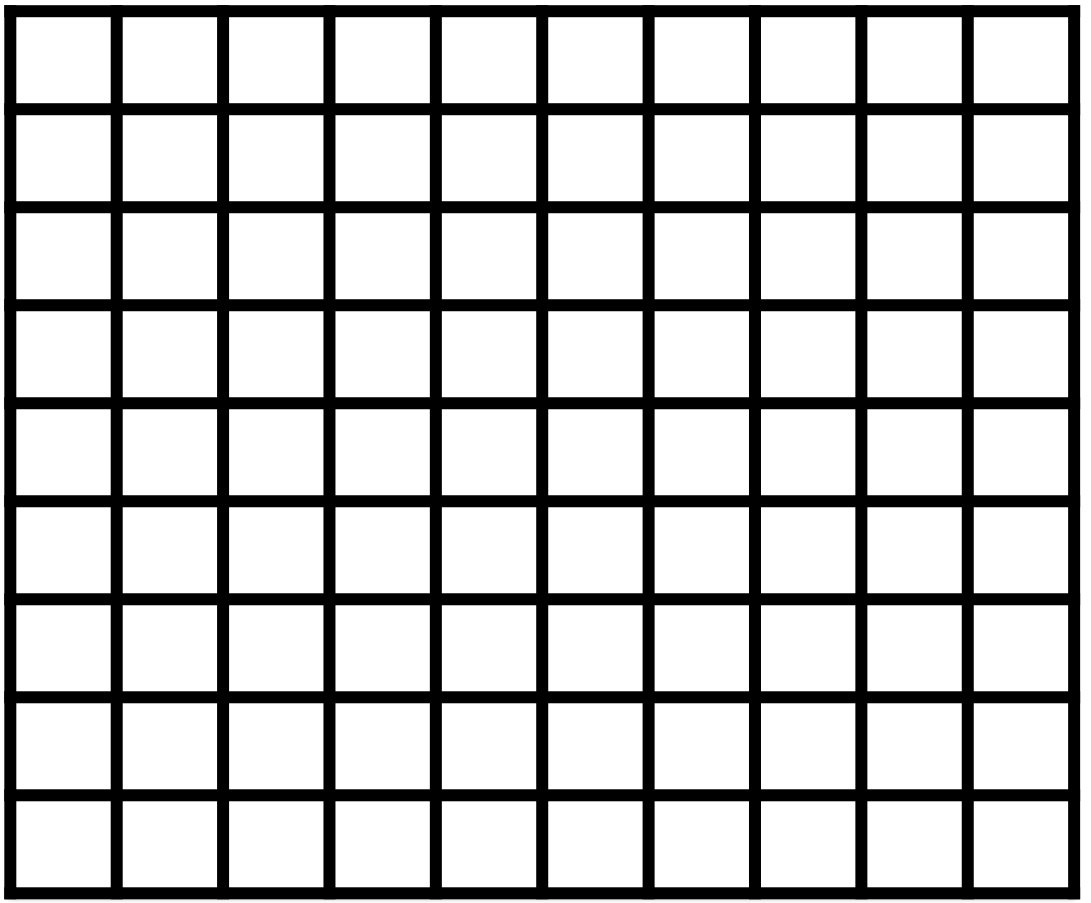
Motivation

- Explore brittelness vs. plasticity of software

- Large quantities of diverse variants
  - Moving target
  - Failure detection



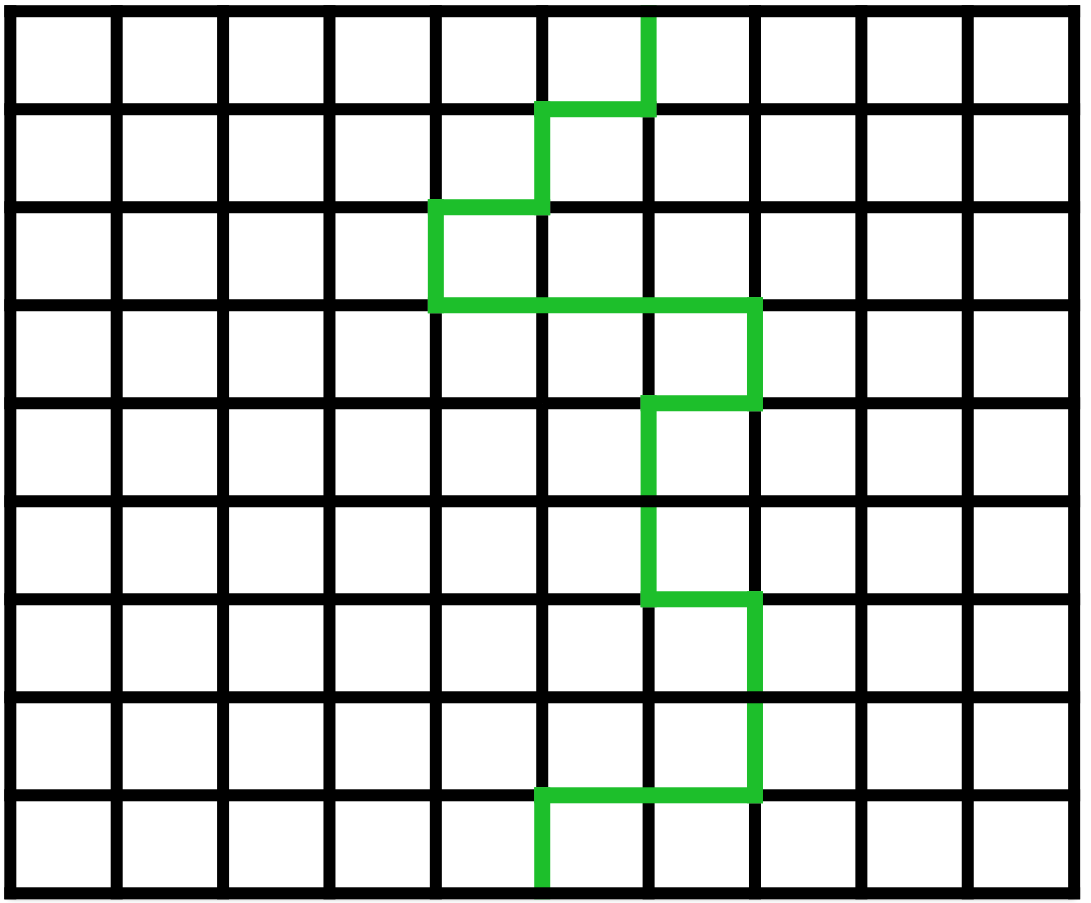
Software brittleness



SRSLSLRSRLLSSRRLRL

G. Berry. « A la chasse aux bugs, la maladie du certain » (8 juin 2011)

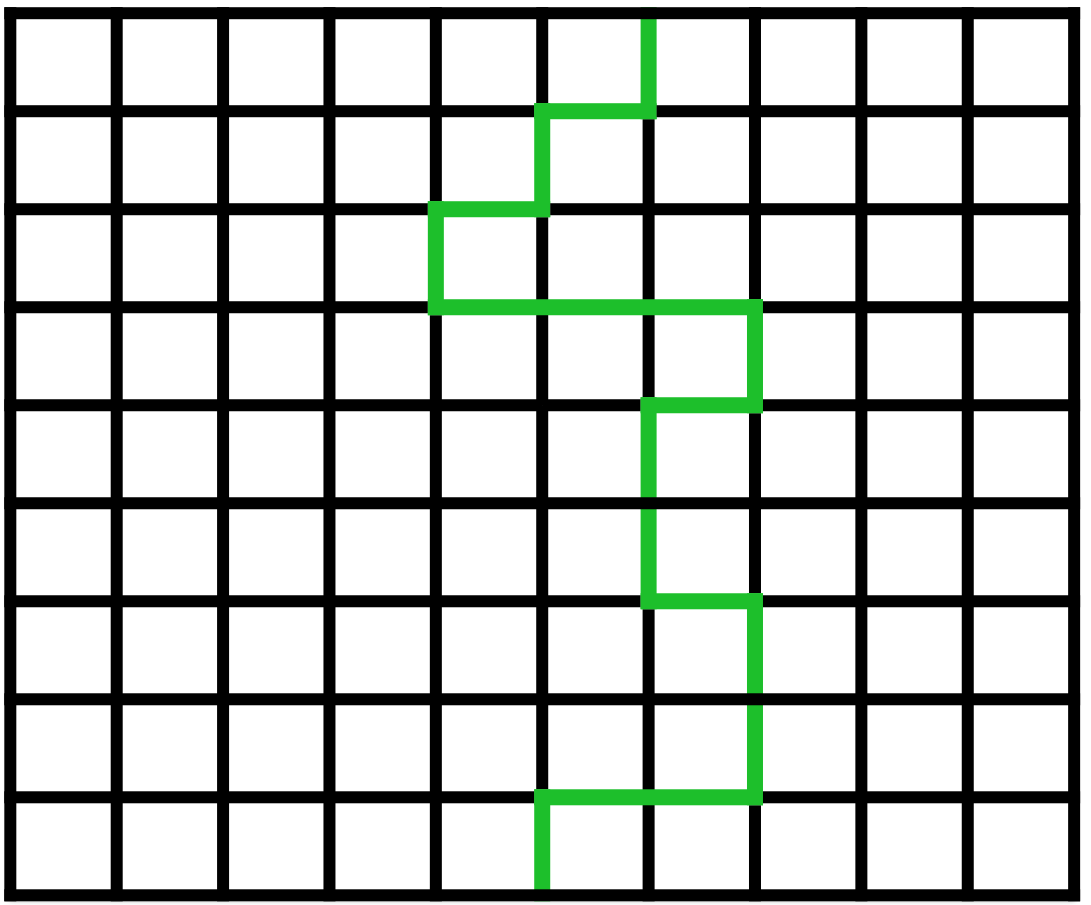
Software brittleness hypothesis



SRSLSLRSRLLSSRRLRL

G. Berry. « A la chasse aux bugs, la maladie du certain » (8 juin 2011)

Software brittleness hypothesis

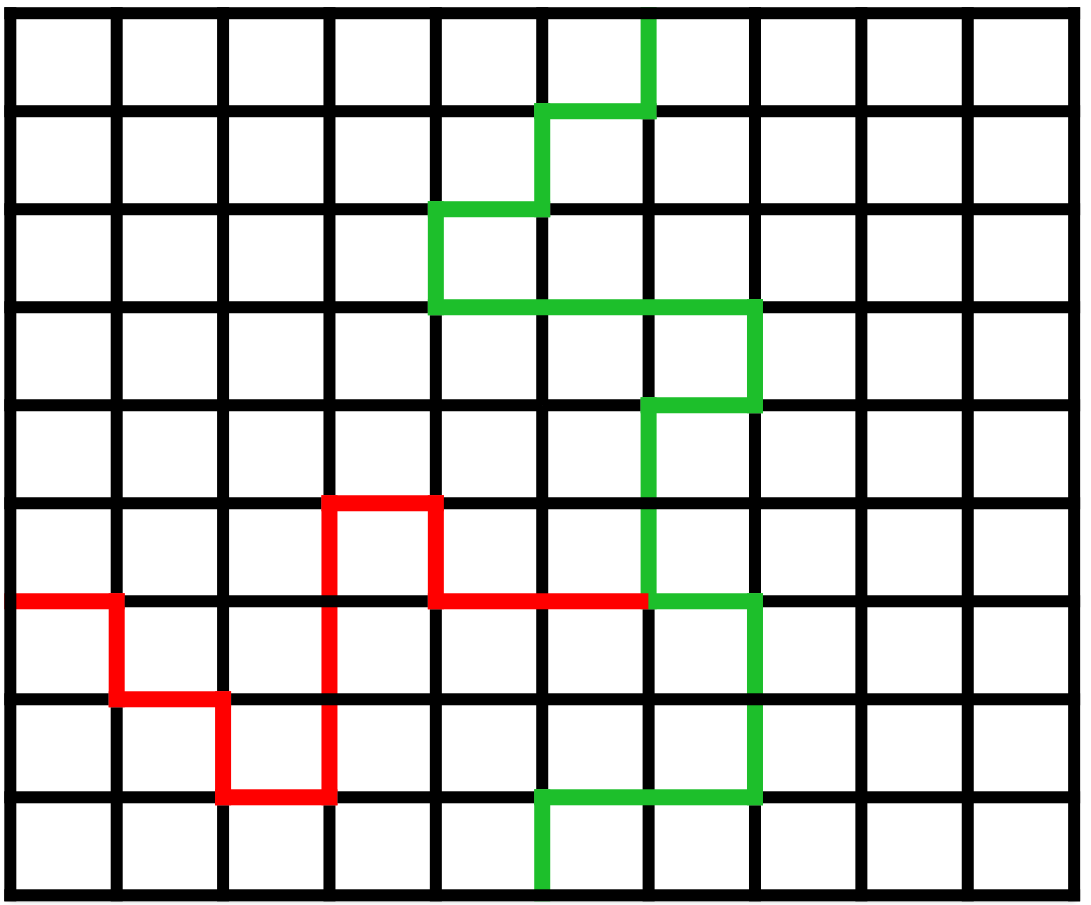


SRSLSLRSRLLSSRRLRL

SRSLSLSSRLLSSRRLRL

G. Berry. « A la chasse aux bugs, la maladie du certain » (8 juin 2011)

Software brittleness hypothesis

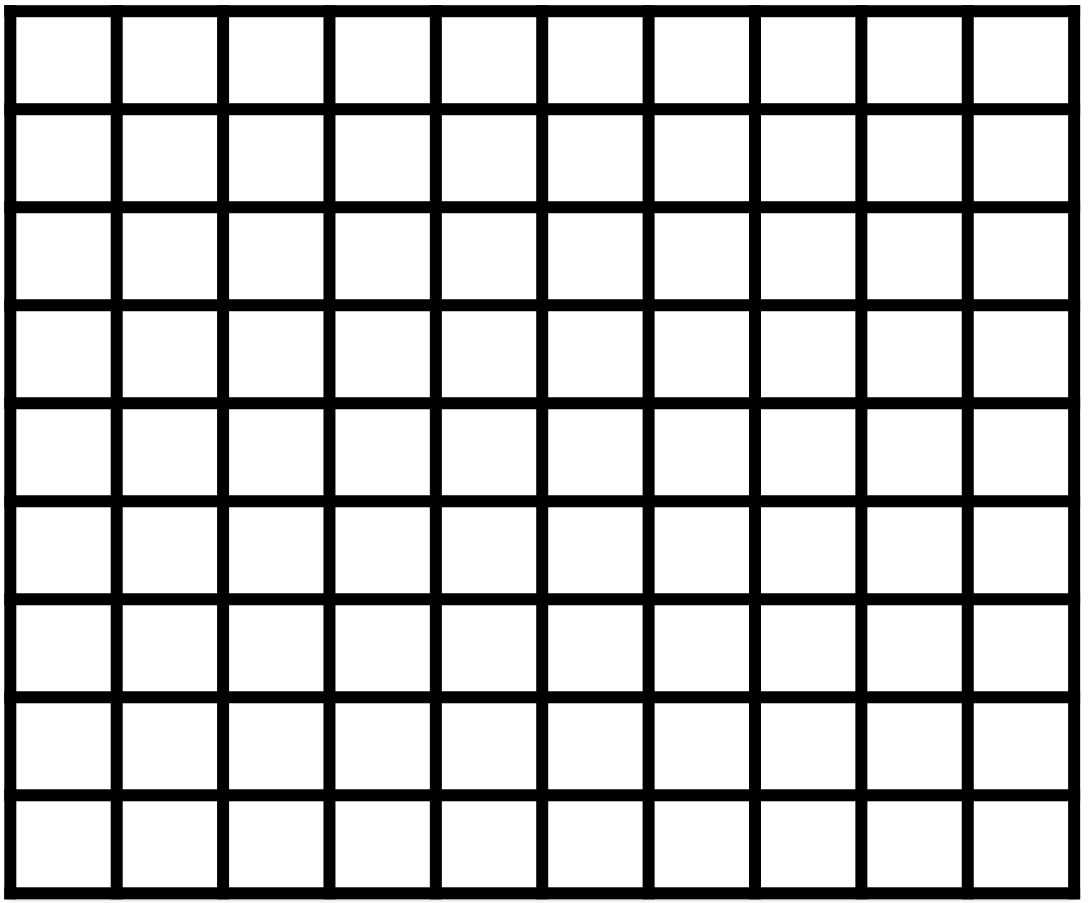


SRSLSLRSRLLSSRRLRL

SRSLSLSSRLLSSRRLRL

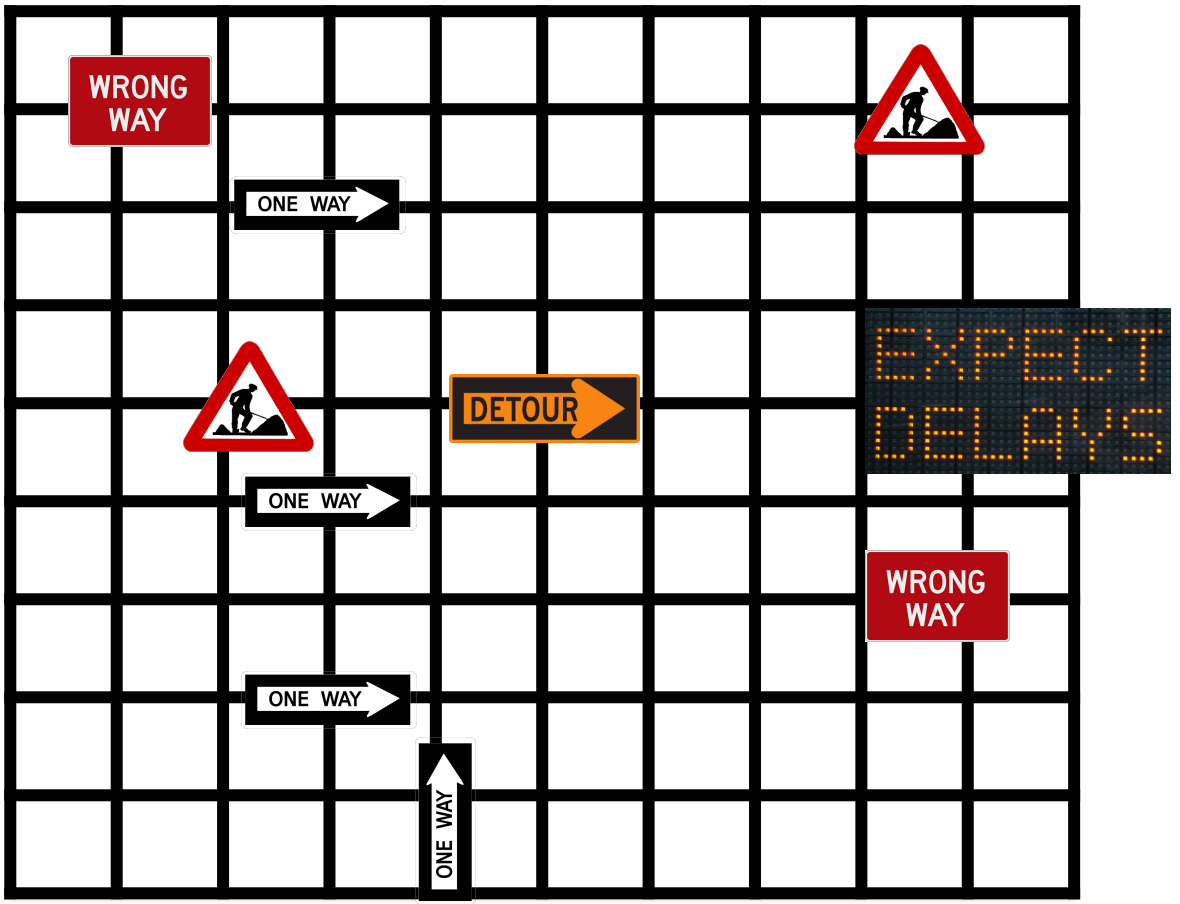
G. Berry. « A la chasse aux bugs, la maladie du certain » (8 juin 2011)

Software brittleness



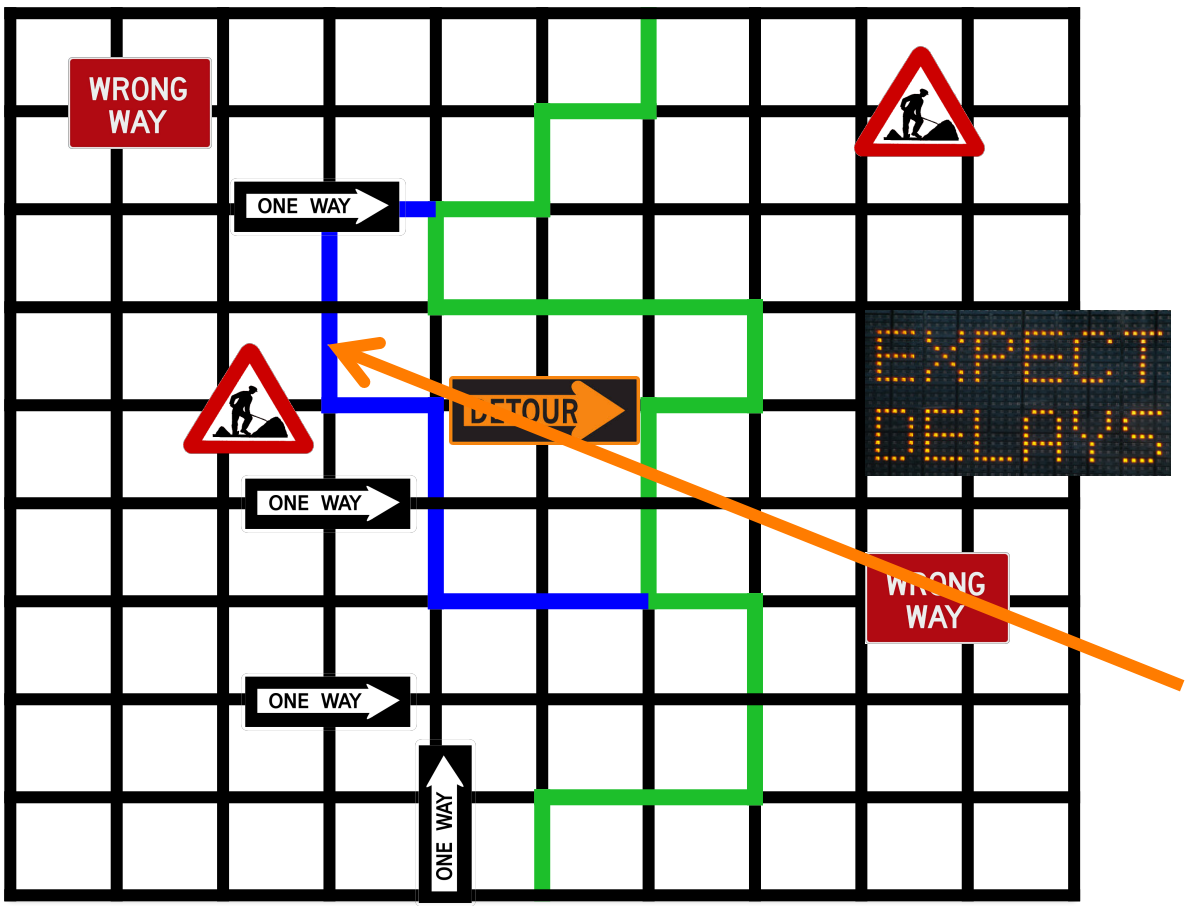

W 46 ST
ONE WAY
NEW
MIRON & SONS
LINEN, INC.
RIGHT LANE
MUST
TURN RIGHT
NYPD
SECURITY
CAMERA
FREE
WIFI
NOW IN
TIMES SQUARE
YAHOO!
NO STOPPING
ANYTIME
DO NOT
ENTER
Office
DEPOT

Software plasticity hypothesis



Software plasticity hypothesis

Rinard et al.
ICSE'10,
FSE'11
POPL'12,
PLDI'14



SRSLSLRSRLLSSRRLRL

SRSLSLSSRLLSSRRLRL

Specification: data and properties

```
fun : Function
assert abs(fun(.5) - 0.25) < 0.05
assert abs(fun(.4) - 0.16) < 0.05
assert abs(fun(.3) - 0.09) < 0.05
```

The test input data specifies the input domain

The assertions specify the level of abstraction

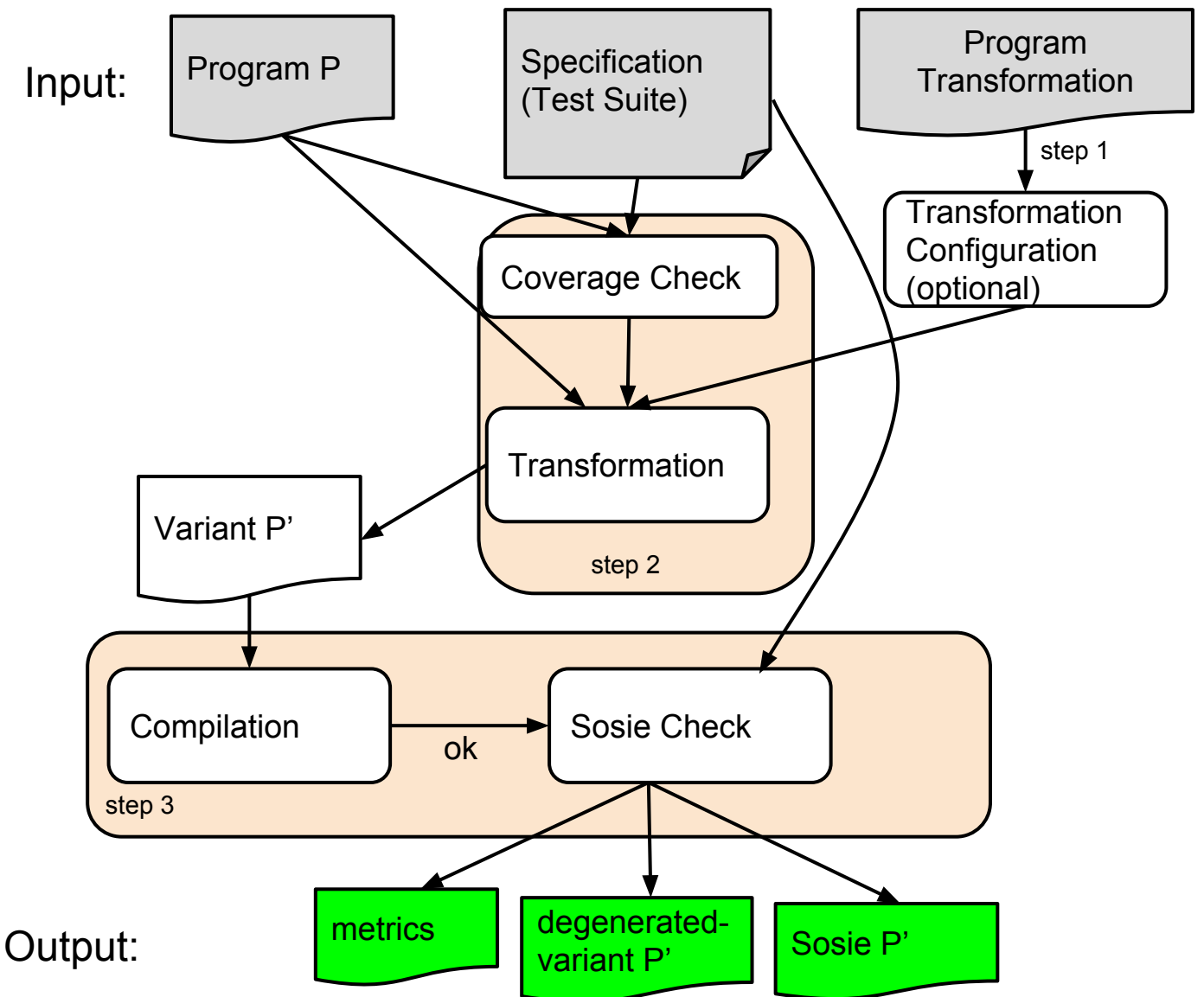
Research questions

Do sosies exist?

Can we automatically synthesize them?

What are effective transformations?

Sosiefication process



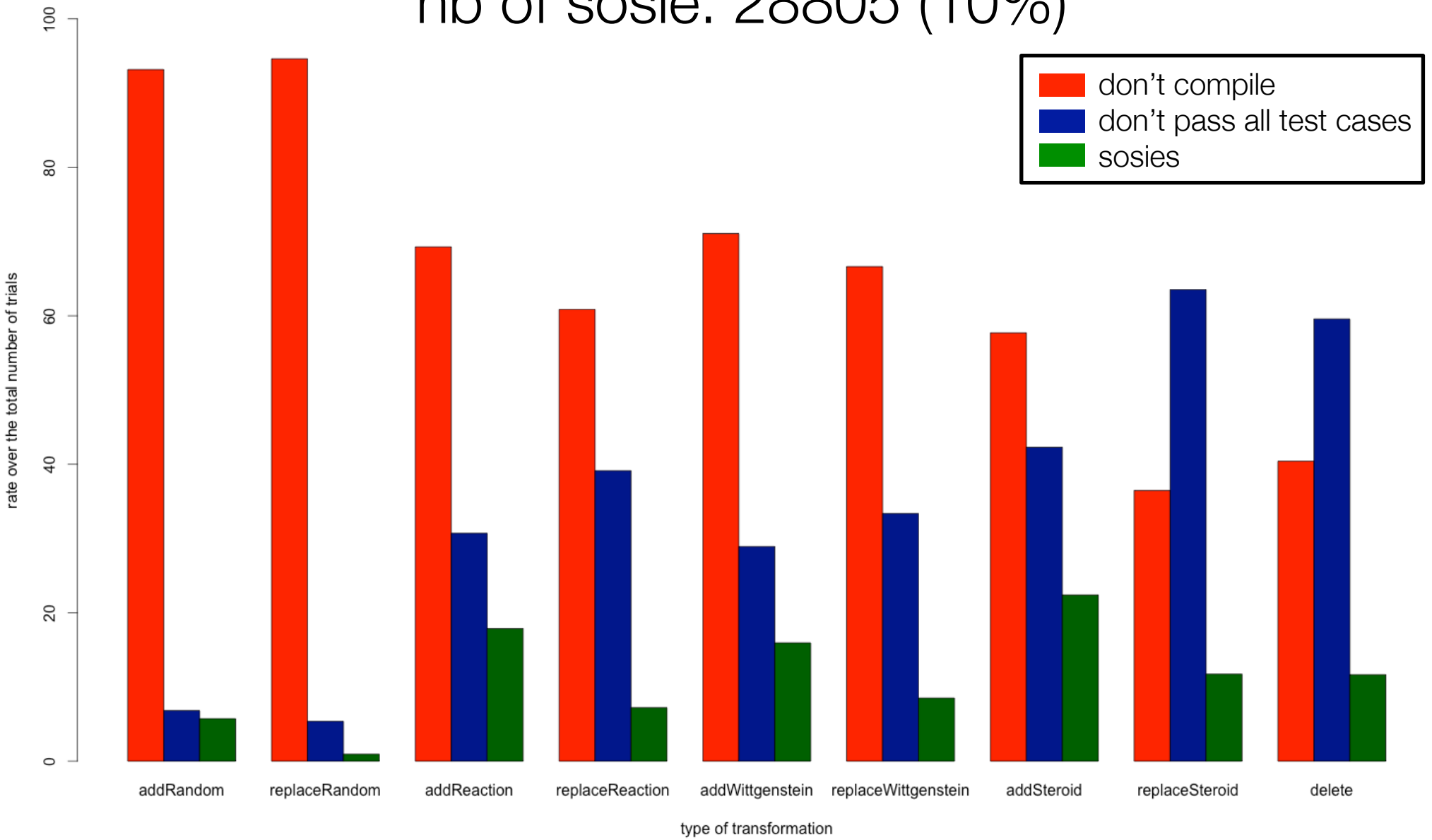
Automatic Synthesis of Sosies

- We add/deleted/replace a given statement by another one and see whether all assertions remain satisfied
  - we pick code from the same program
- Four strategies
  - random
  - wittgenstein: replace with variables that have the same name
  - reaction: replace with variables that have the same type
  - steroid: reaction + rename variables

Experimental data

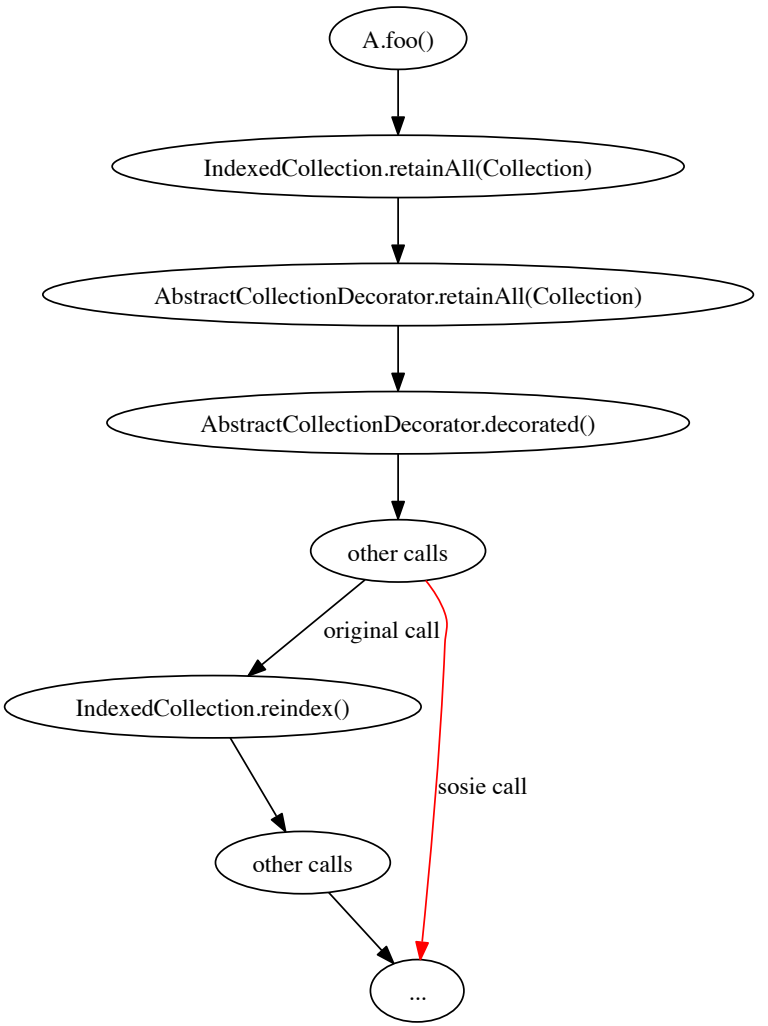
	#test cases	#assert	coverage	#statement	compile time	test time
Junit	721	1535	82%	2914	4.5	14.4
EasyMock	617	924	91%	2042	4	7.8
Dagger (core)	128	210	85%	674	5.1	11.2
JBehave-core	485	1451	89%	4984	5.5	22.9
Metrics	214	312	79%	1471	4.7	7.7
commons-collections	1121	5397	84%	9893	7.9	22.9
commons-lang	2359	13681	94%	11715	6.3	24.6
commons-math	3544	9559	92%	47065	9.2	144.2
clojure	NA	NA	71%	18533	105.1	185

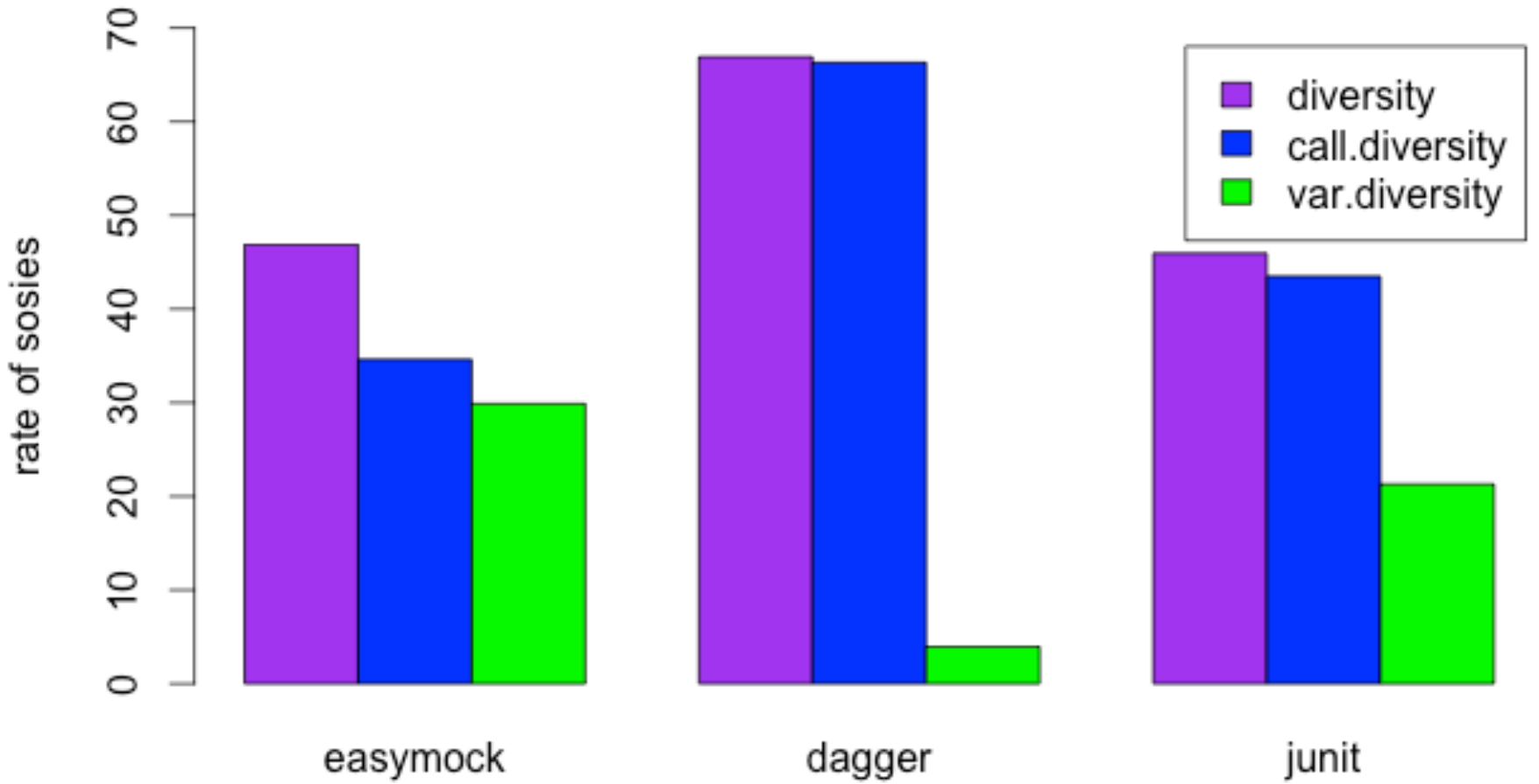
nb of trial: 298938
nb of compile: 81394
nb of sosie: 28805 (10%)



Computation diversity

- Goal: unpredictability of execution flow
- Computation monitoring:
  - method calls diversity
  - variable diversity





Easymock: 465 sosies
Dagger: 481 sosies
Junit: 446 sosies

Conclusion

- Sosies exist
  - for all programs
- Sosies can exhibit computation diversity
- Next steps
  - variability-aware execution
  - is computational diversity unbounded?

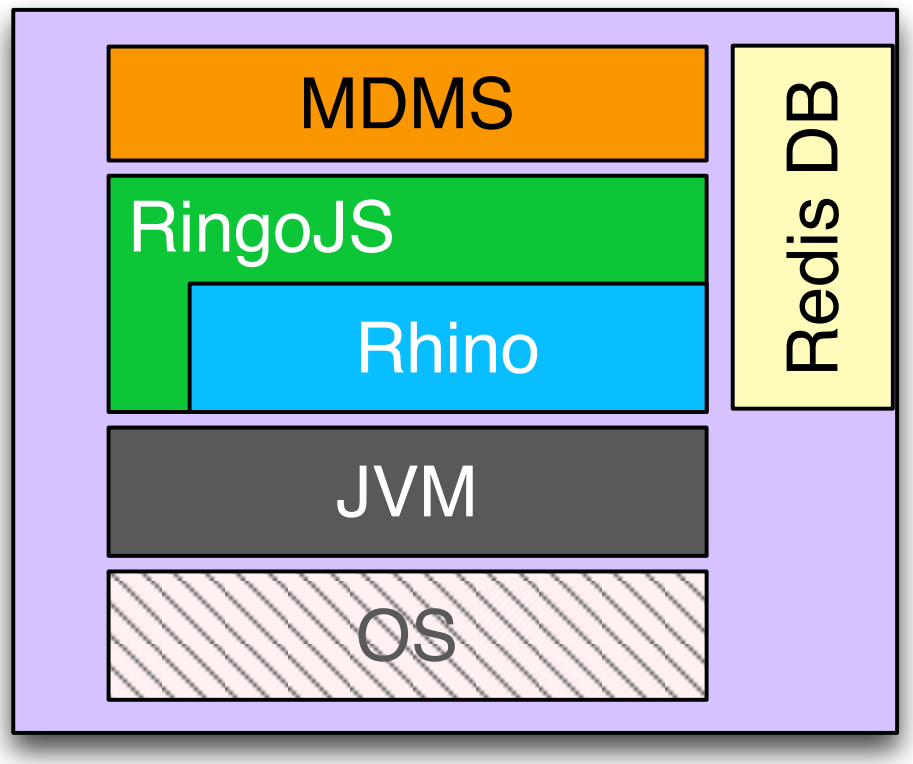
https://github.com/DIVERSIFY-project/sosies-generator
http://diversify-project.eu/sosiefied-programs/

References

- Zeyuan Allen Zhu, Sasa Misailovic, Jonathan A. Kelner, Martin C. Rinard: Randomized accuracy-aware program transformations for efficient approximate computations. POPL 2012: 441-454
- Eric Schulte, Jonathan Dorn, Stephen Harding, Stephanie Forrest, Westley Weimer: Post-compiler software optimization for reducing energy. ASPLOS 2014: 639-652
- Frederick B Cohen: Operating system protection through program evolution. Computers & Security 12, 6 (1993): 565–584.

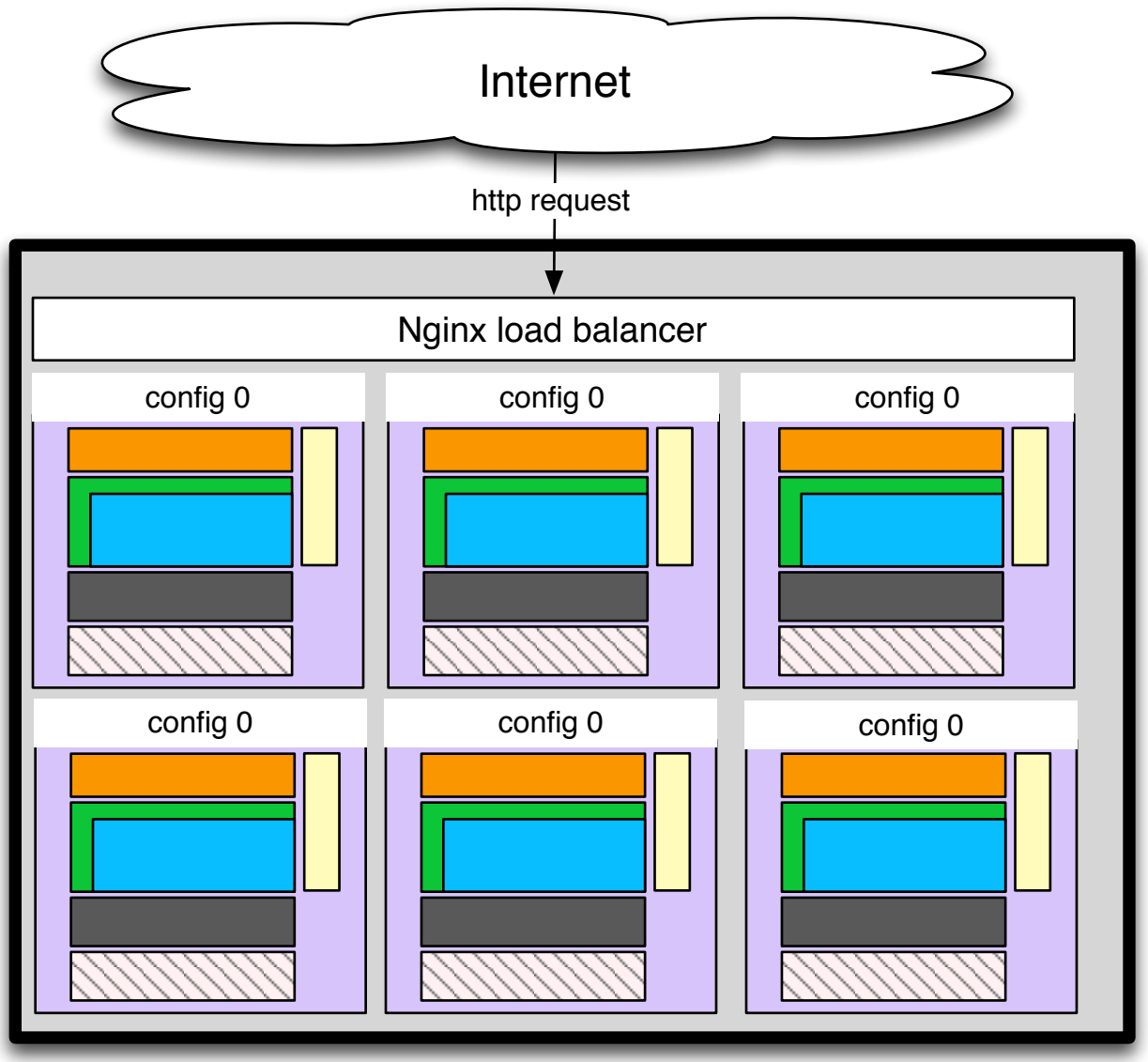
Sosies on line

- MDMS
  - simple blog app
  - JS on client and server sides
- Server side stack
  - JS
  - Java
  - DB
  - environment



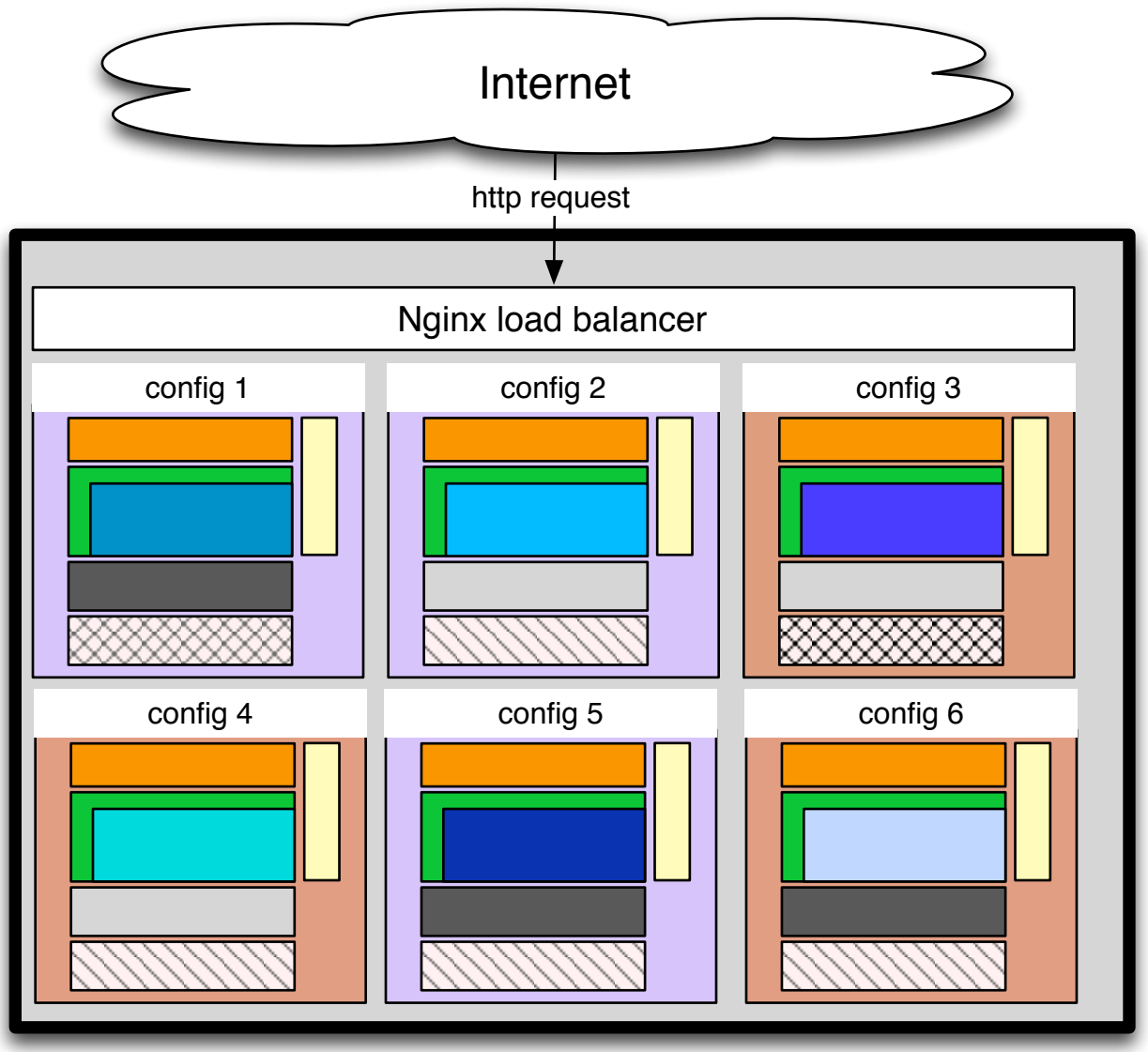
Sosies on line

- Monoculture
  - multiple instances for performance
  - load balancer
  - all instances are clones

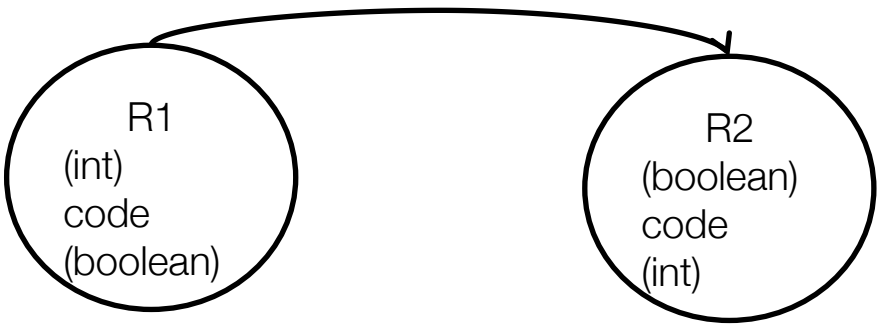


Sosies on line

- Diversified deployment
  - All server instances are different
  - Combine natural and artificial diversity

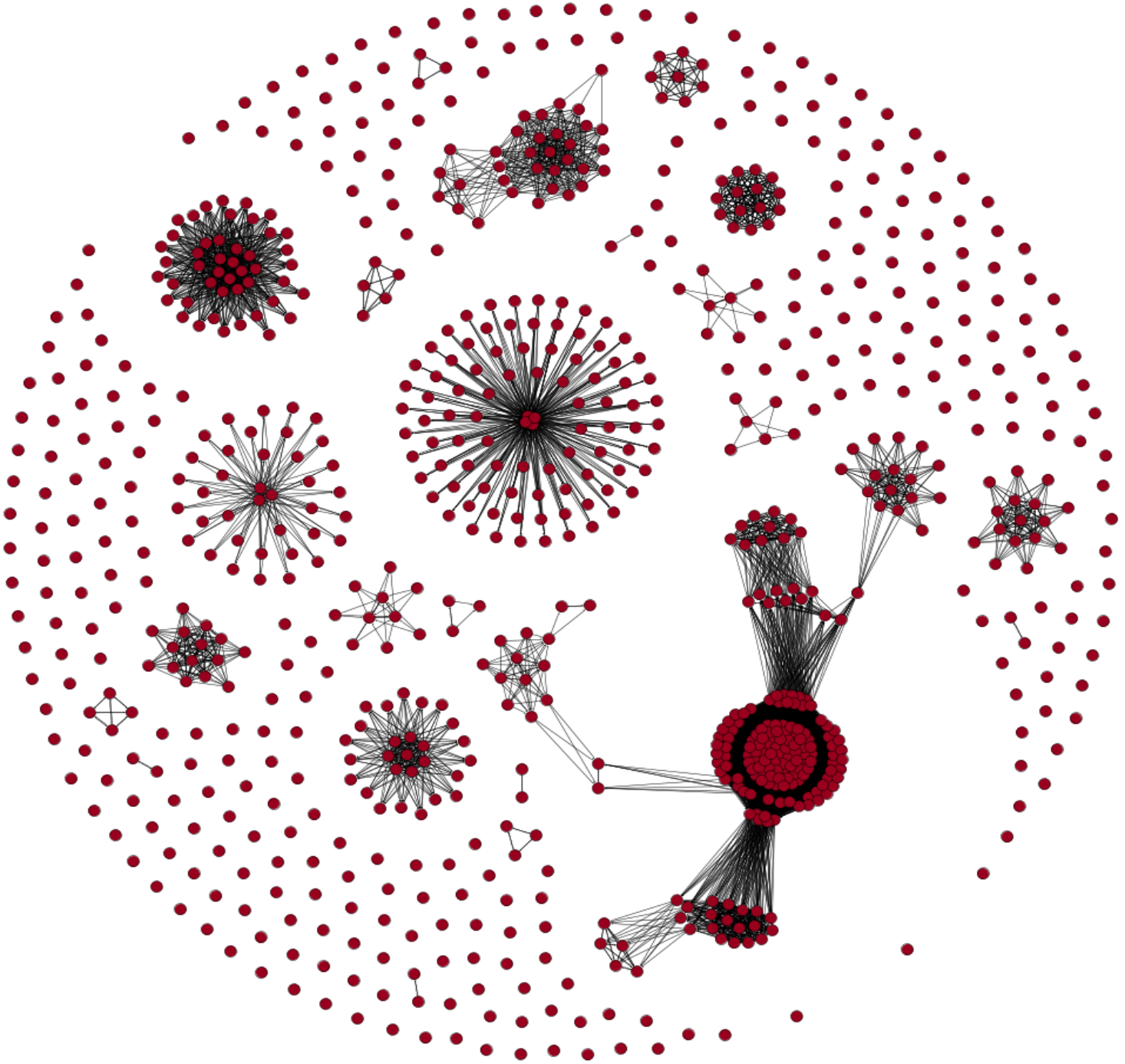


Reactions graph



- Reactions graph
  - one node per reaction
  - there is an edge between n1 and n2 if

n2.in_context == n1.in_context ⋁ n1.out_context



Two reactions graph (apache.common)

- Statement reactions graph
  - #edges = 12304
  - #nodes = 863
  - graph-diameter = 3
  - avg path length = 1.466
  - avg degree = 14.257
- Expression reactions graph
  - #edges = 37650
  - #nodes = 1953
  - graph-diameter = 4
  - avg path length = 1.162
  - avg degree = 19.278