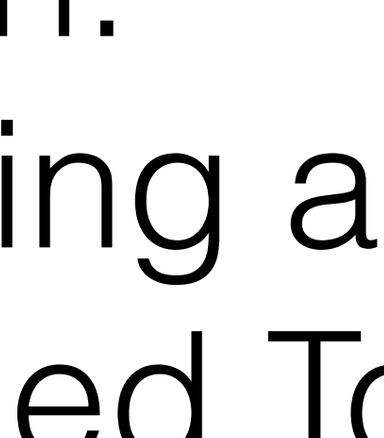
SpaceSearch: A Library for Building and Verifying Solver-Aided Tools



Konstantin Weitz

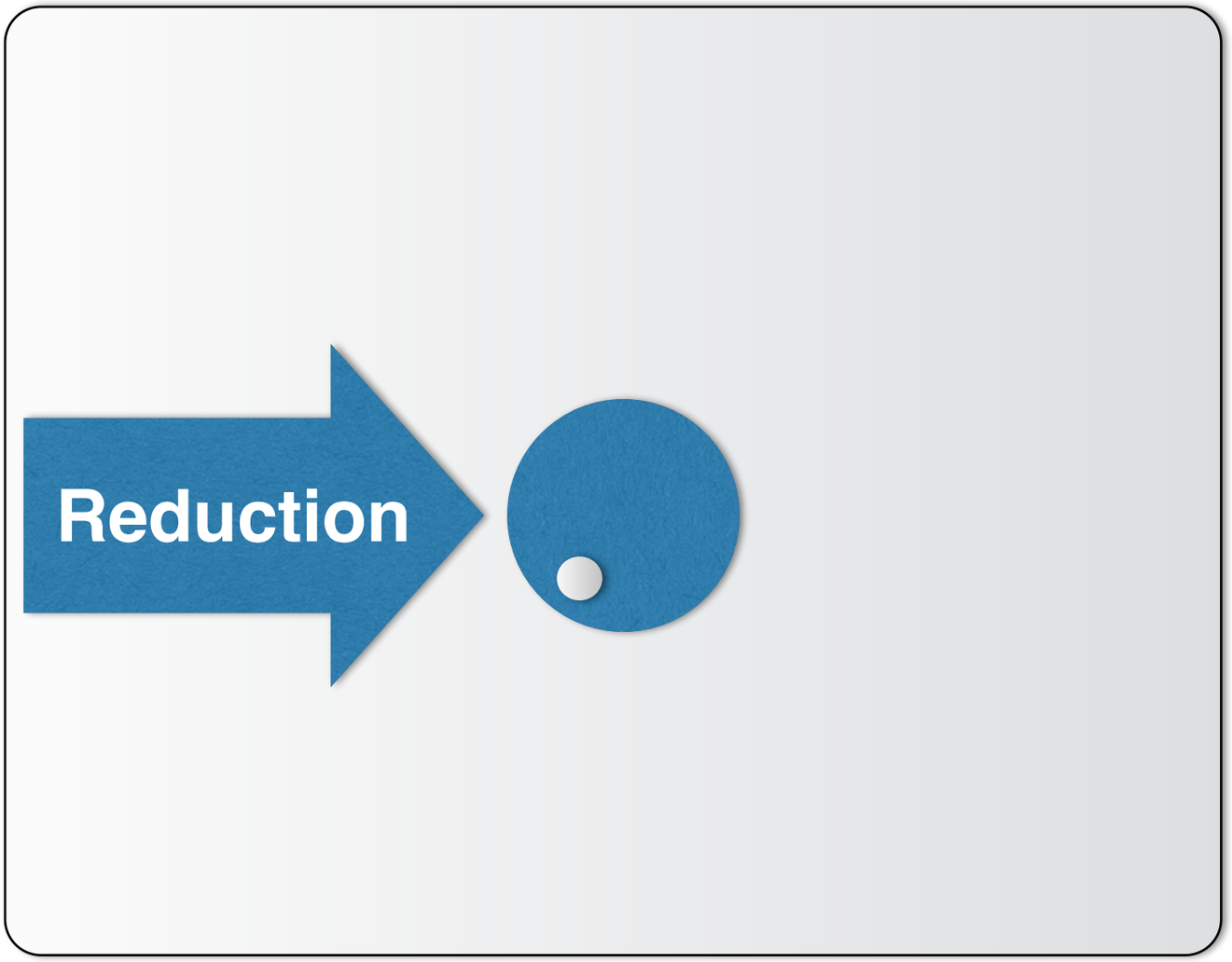
Steven S. Lyubomirsky

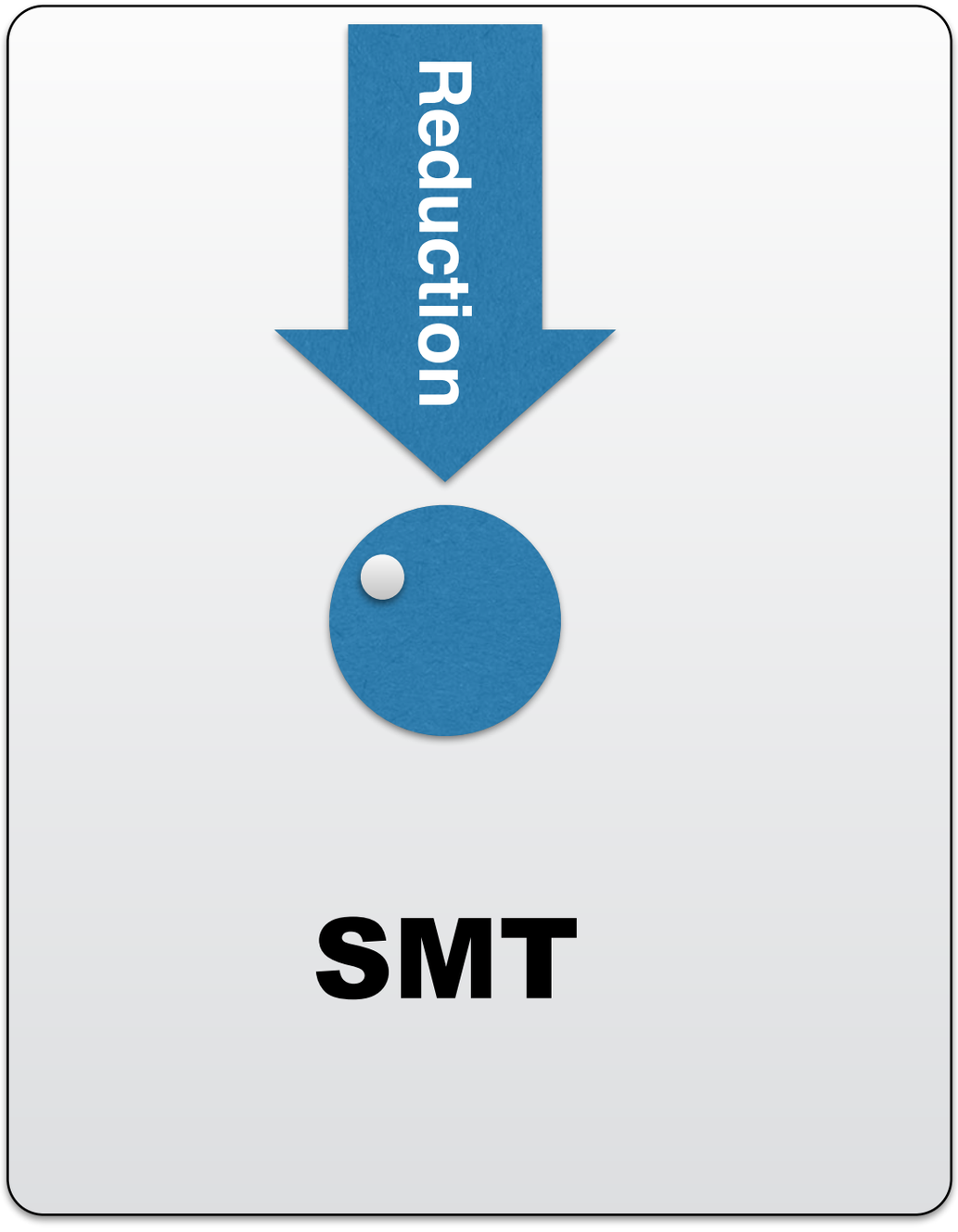
Stefan Heule

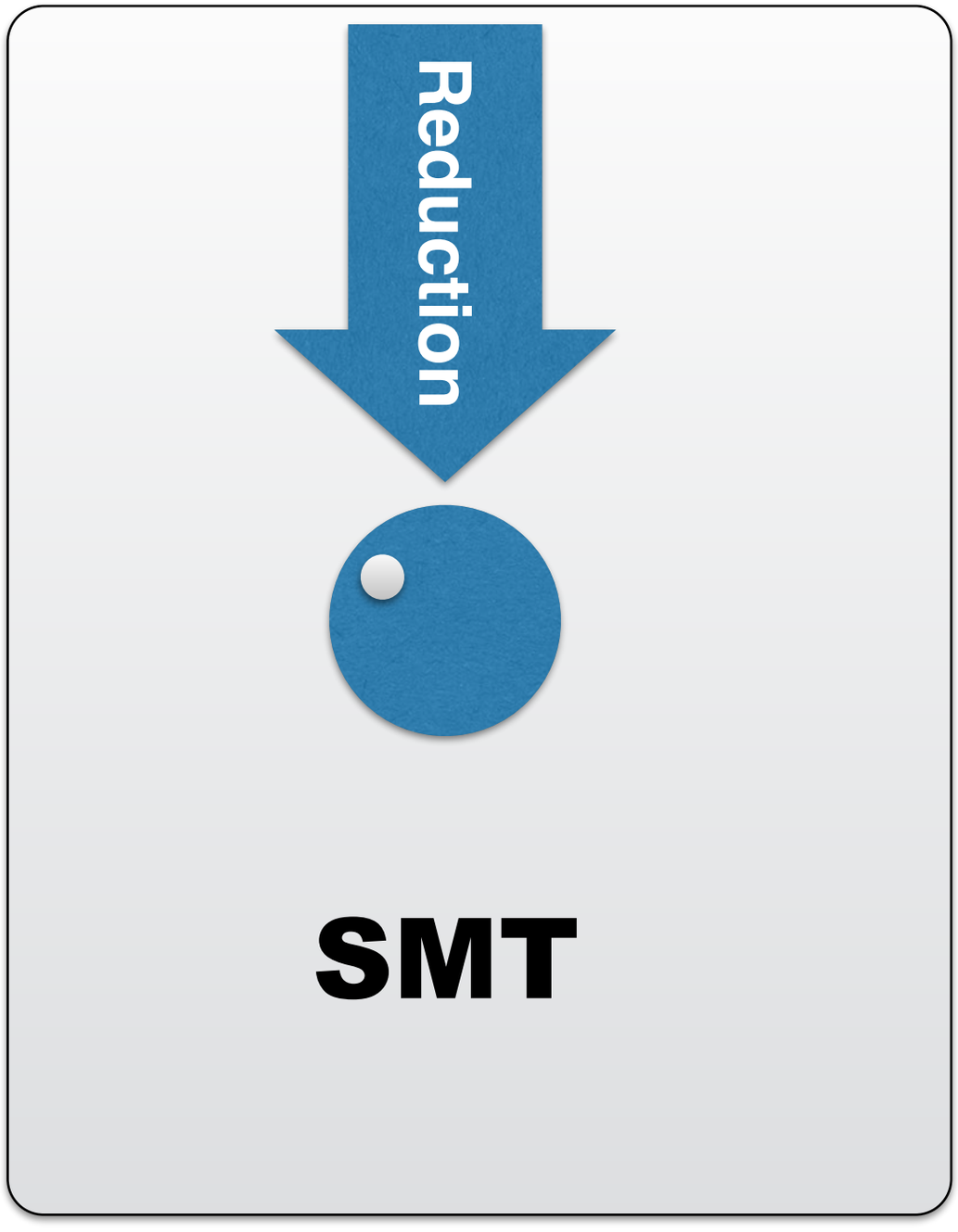
Emina Torlak

Michael D. Ernst

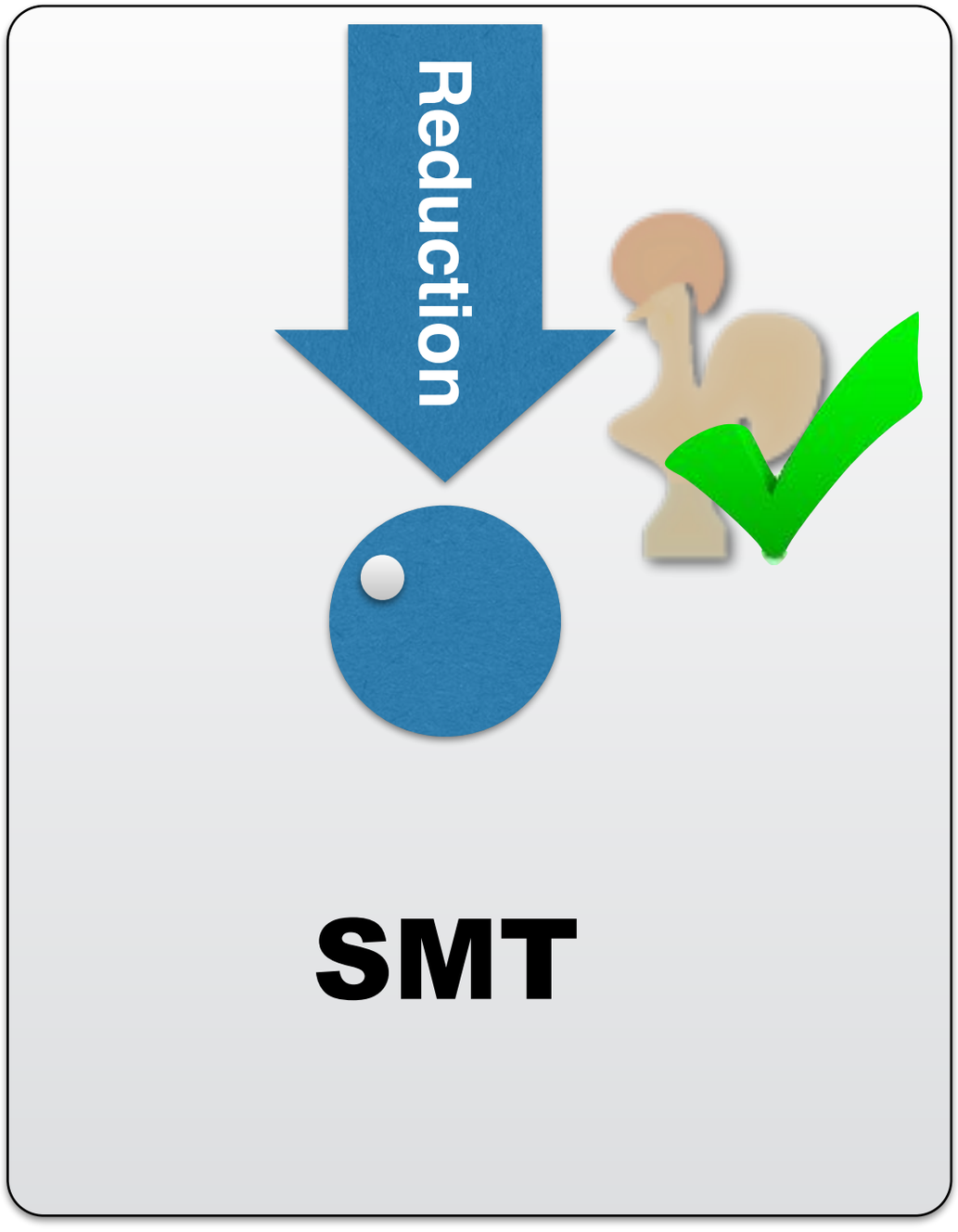
Zachary Tatlock


Reduction
?

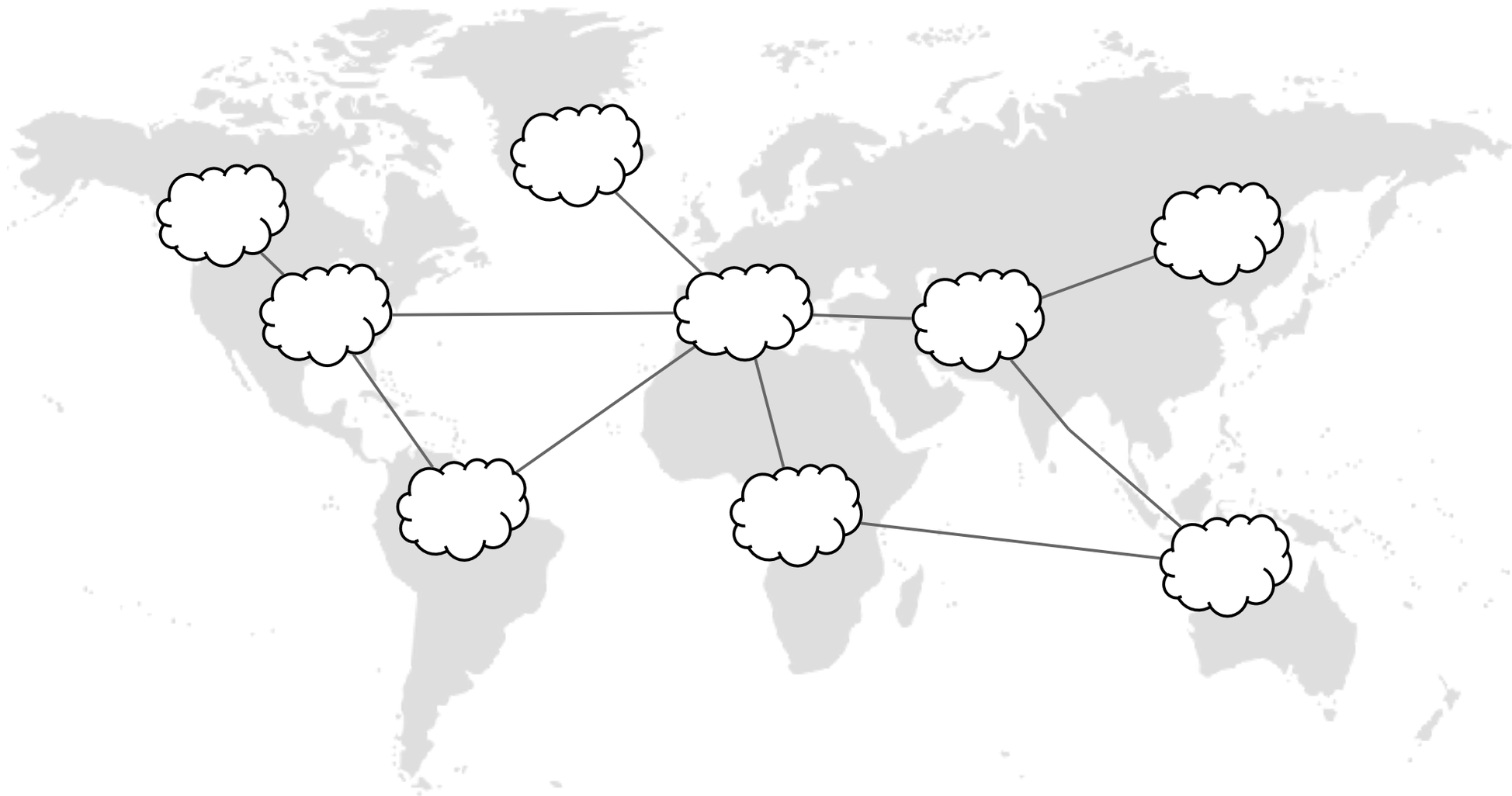

Reduction
SMT


Reduction
SMT

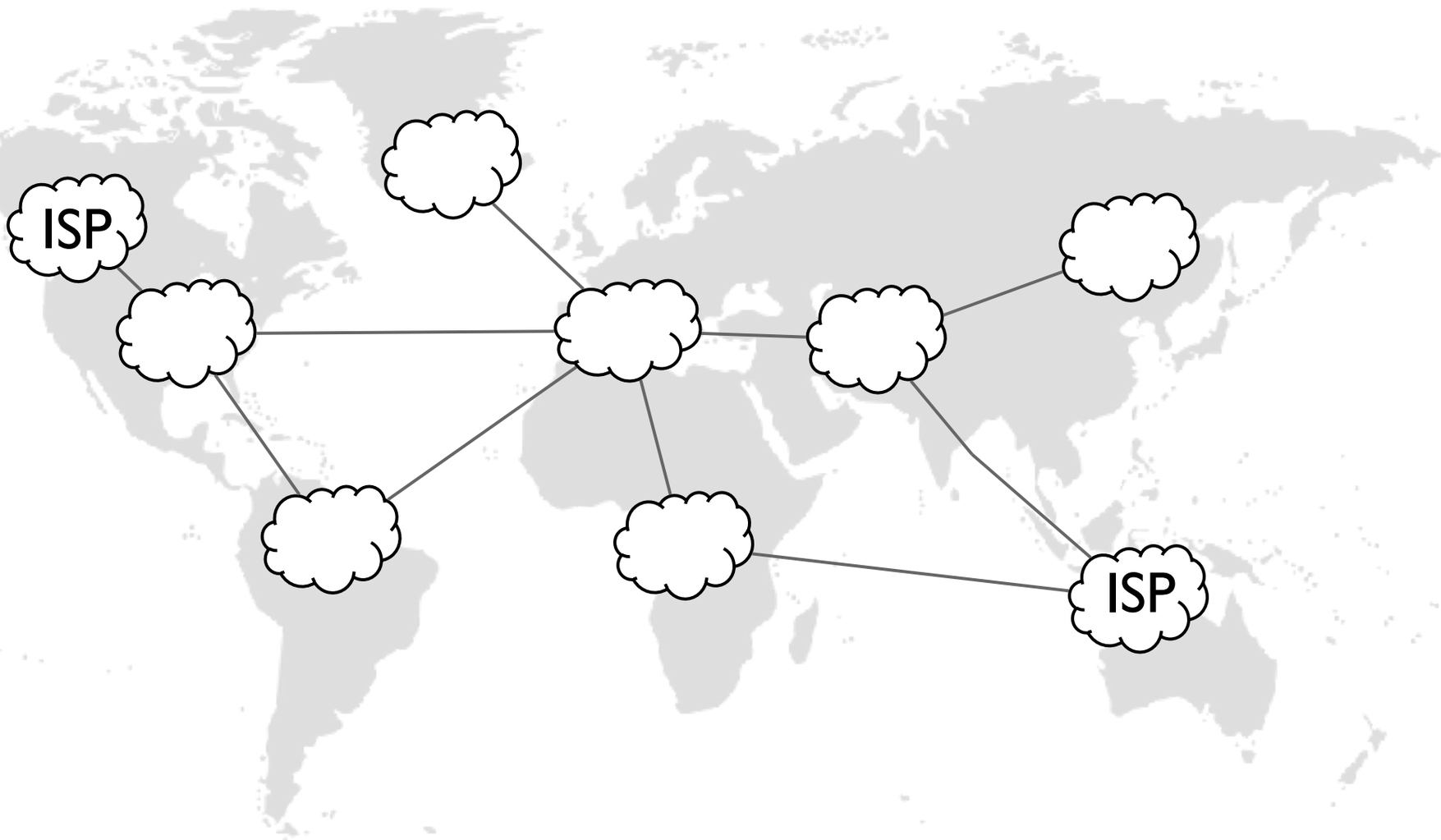
SpaceSearch



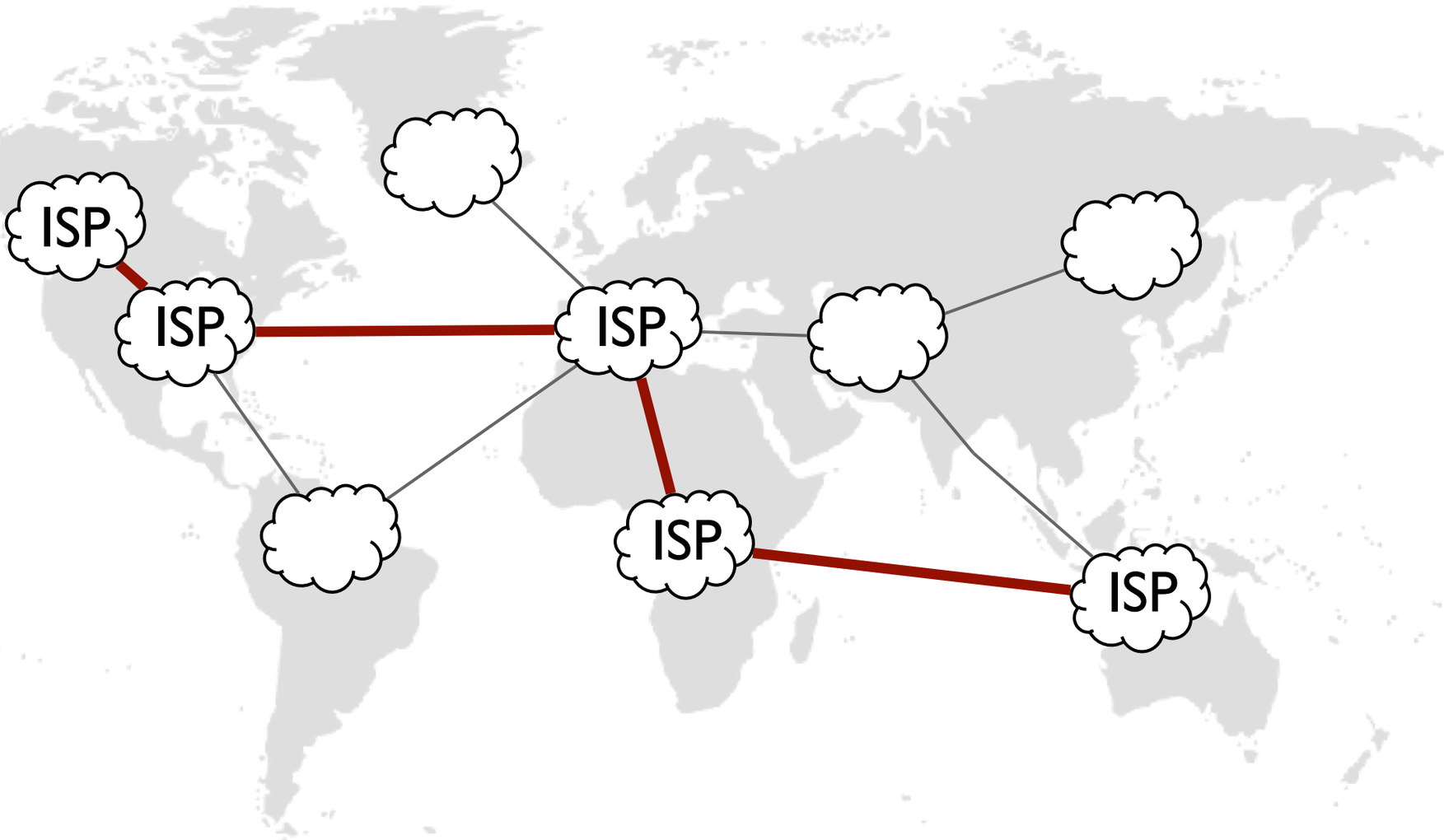
The Border Gateway Protocol



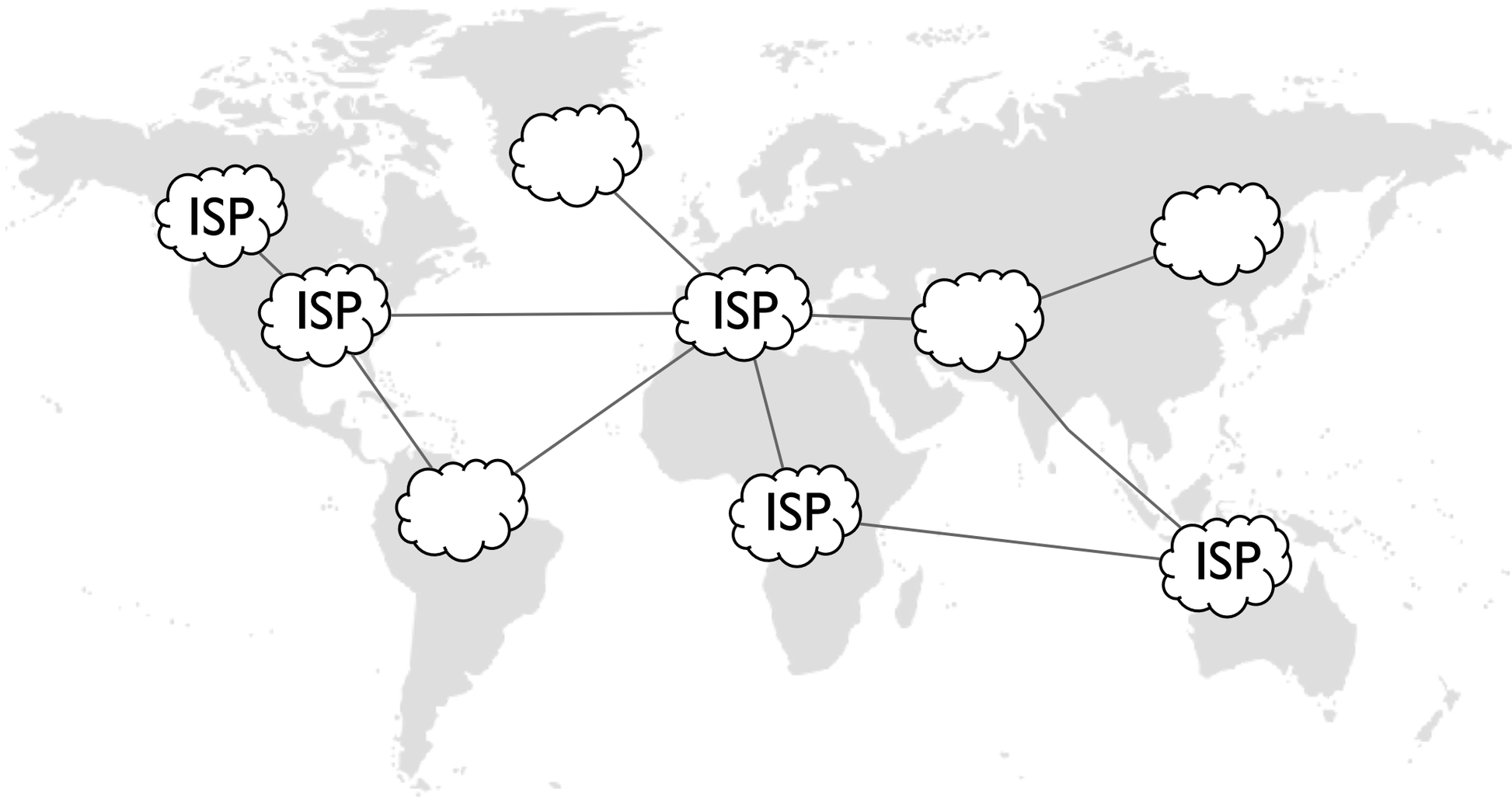
The Border Gateway Protocol



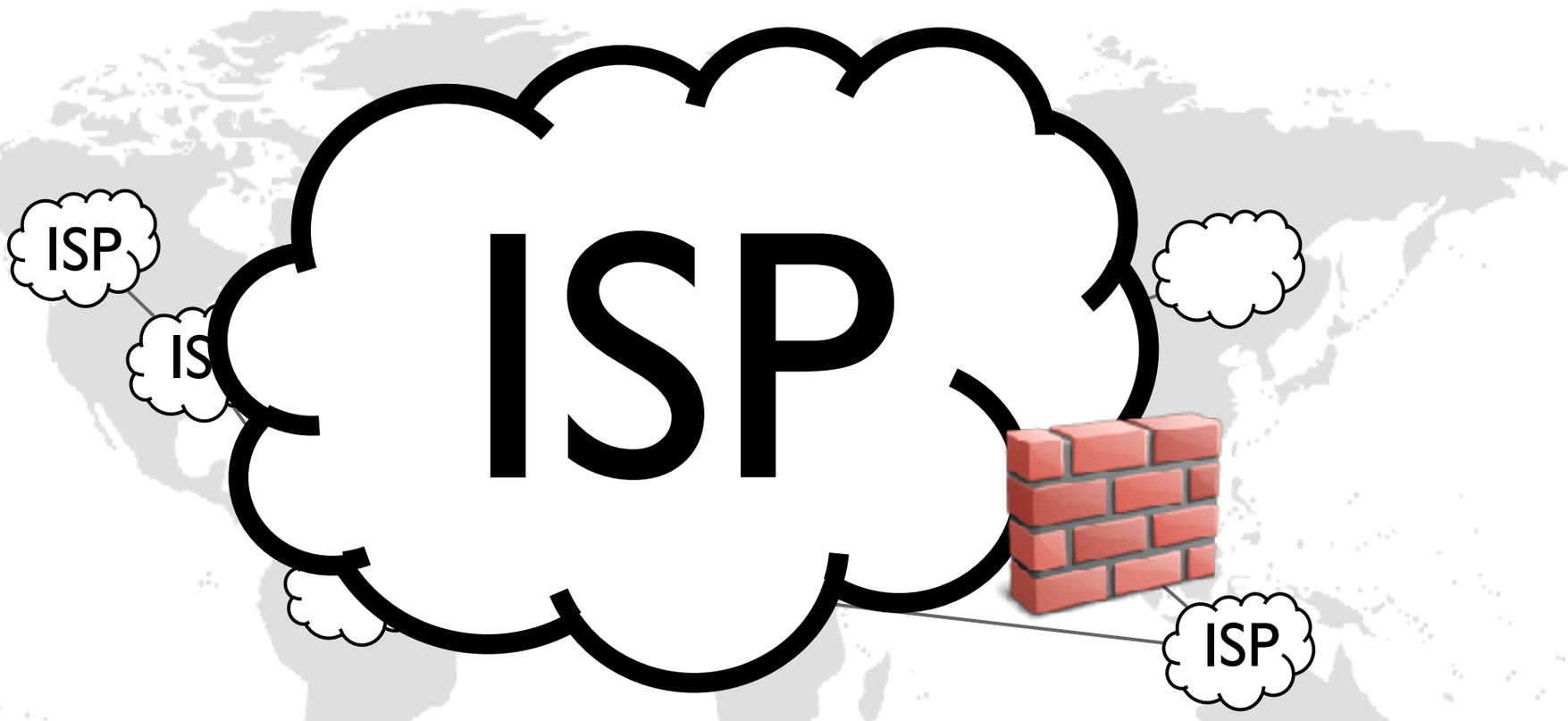
The Border Gateway Protocol



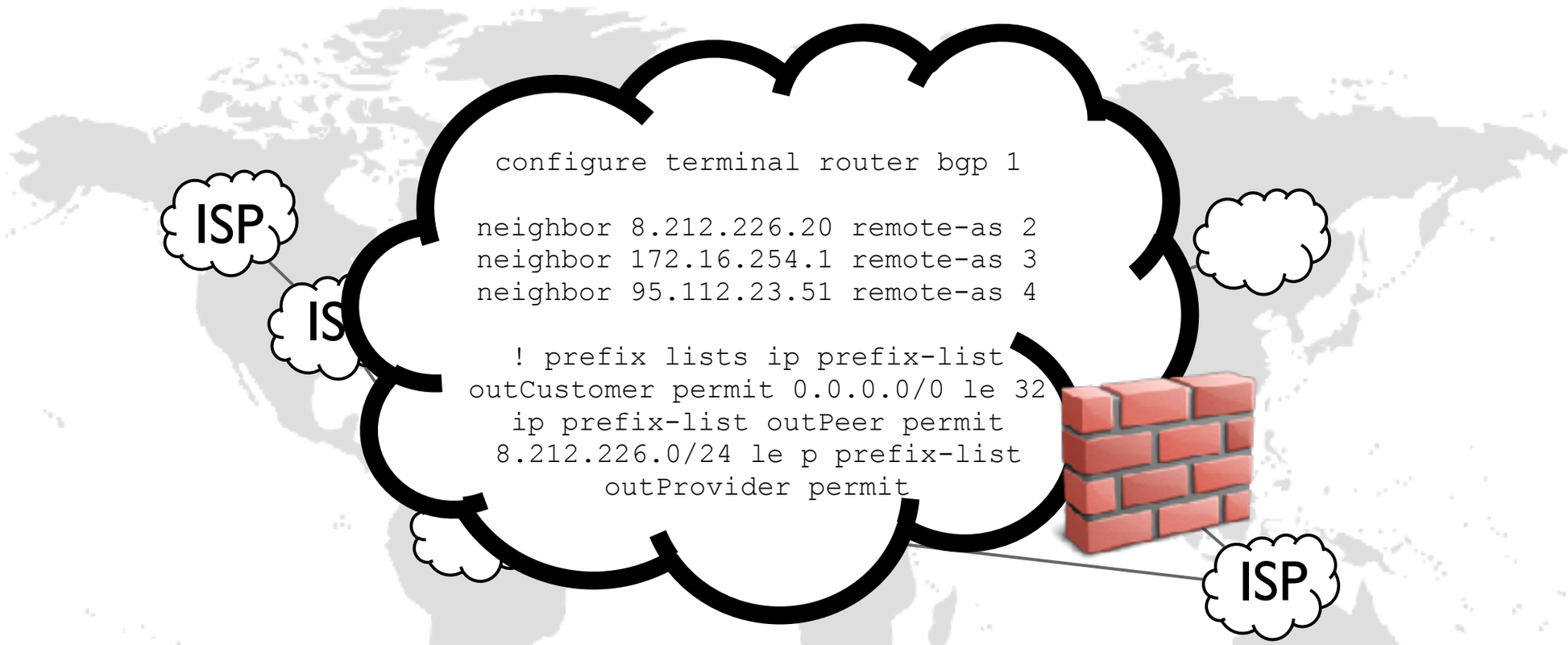
The Border Gateway Protocol



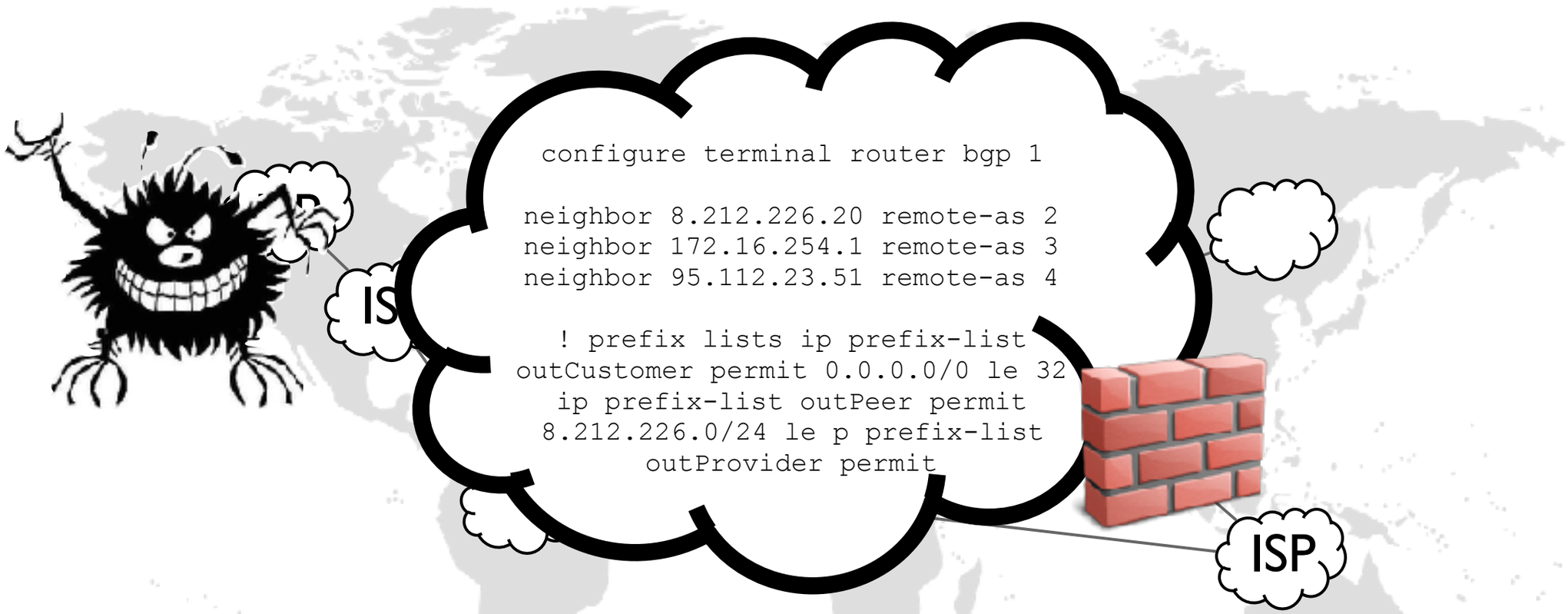
The Border Gateway Protocol



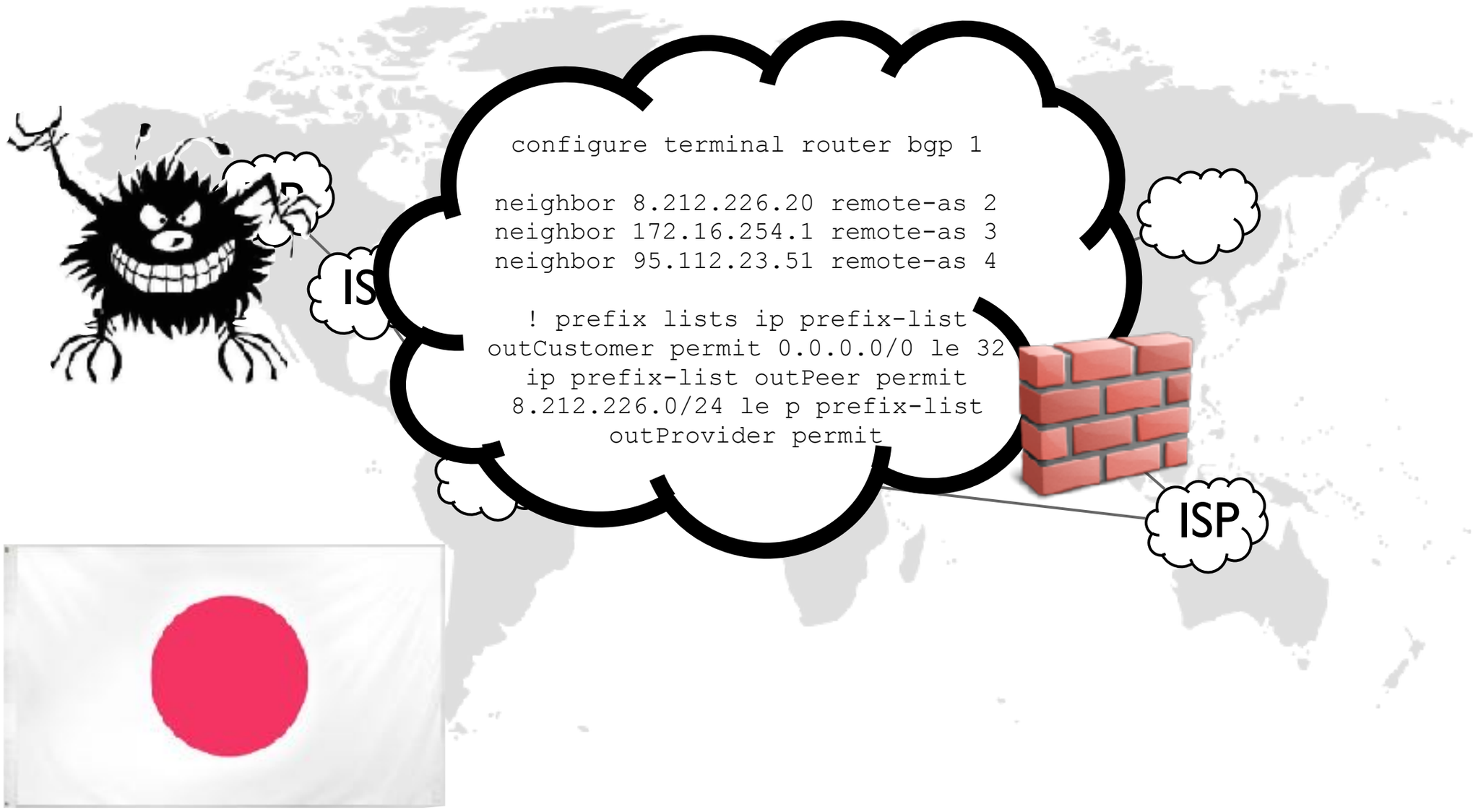
The Border Gateway Protocol



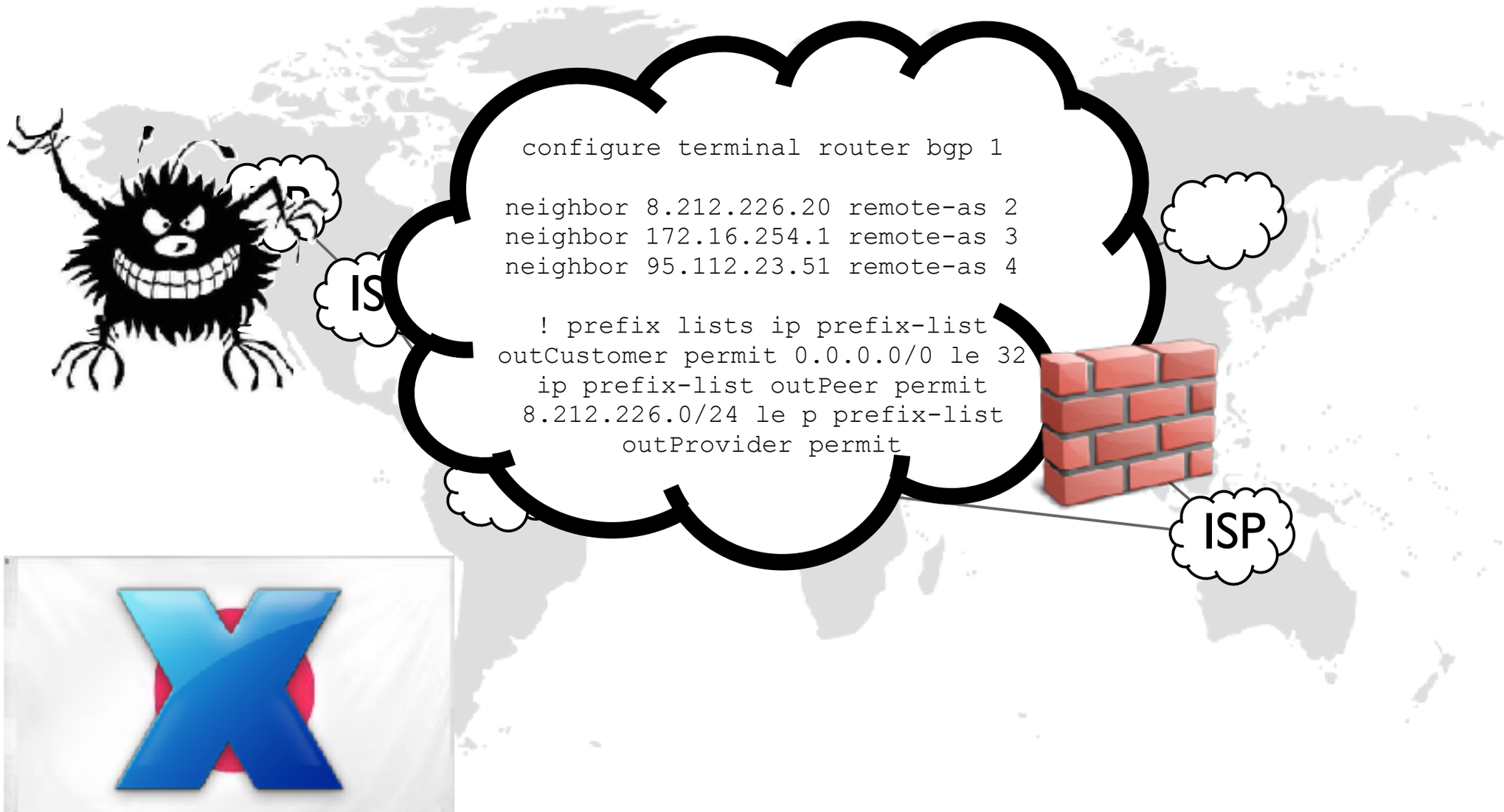
The Border Gateway Protocol



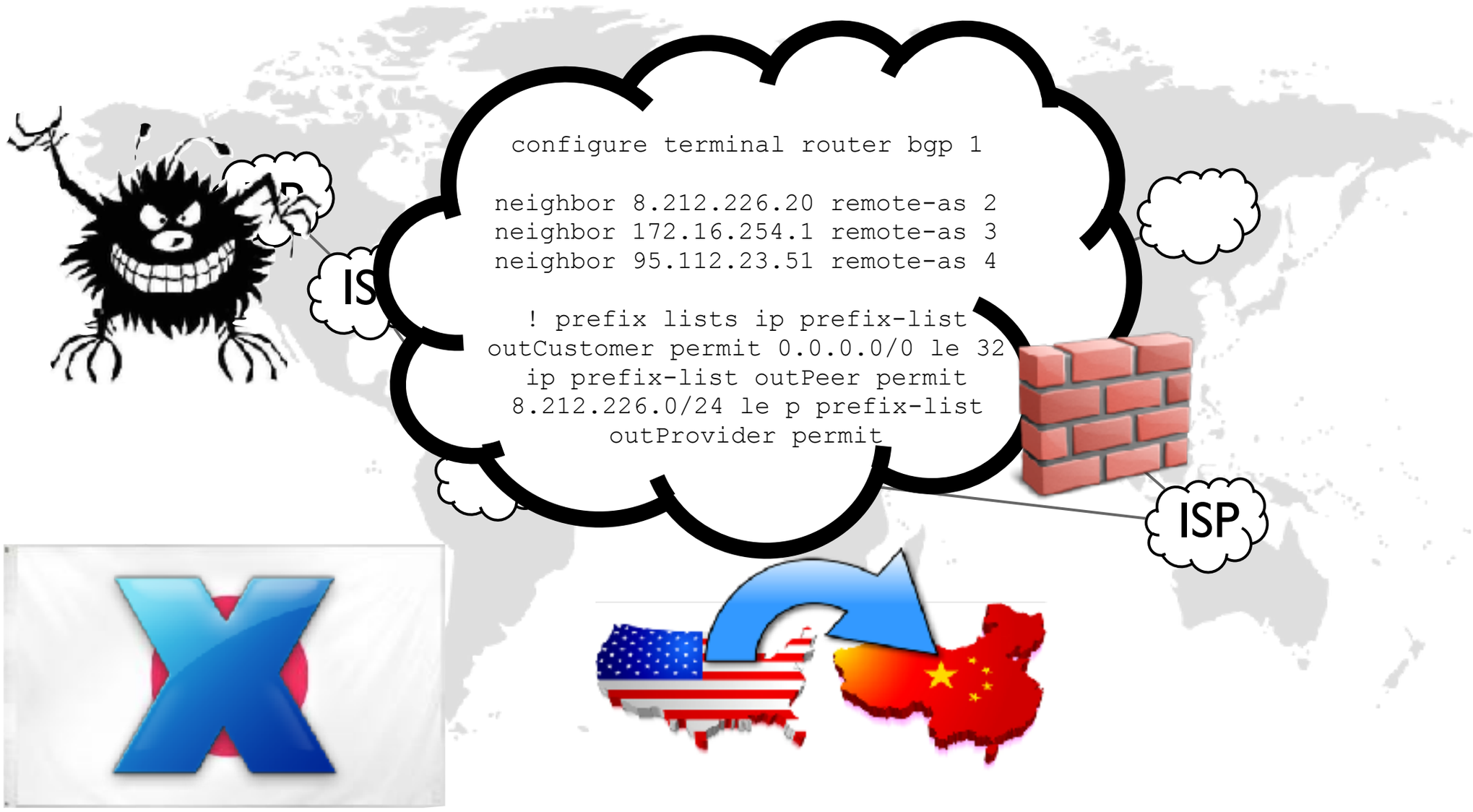
The Border Gateway Protocol



The Border Gateway Protocol



The Border Gateway Protocol

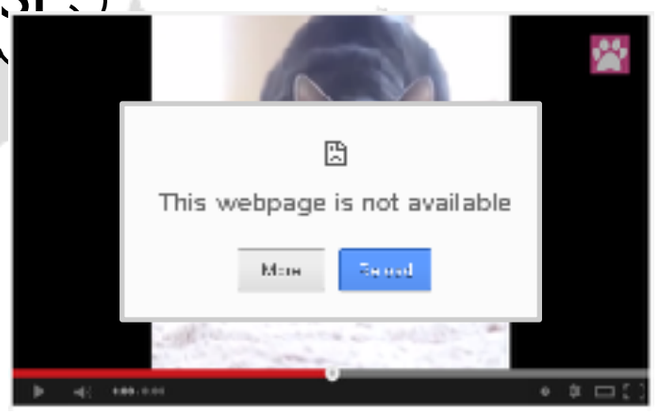


The Border Gateway Protocol

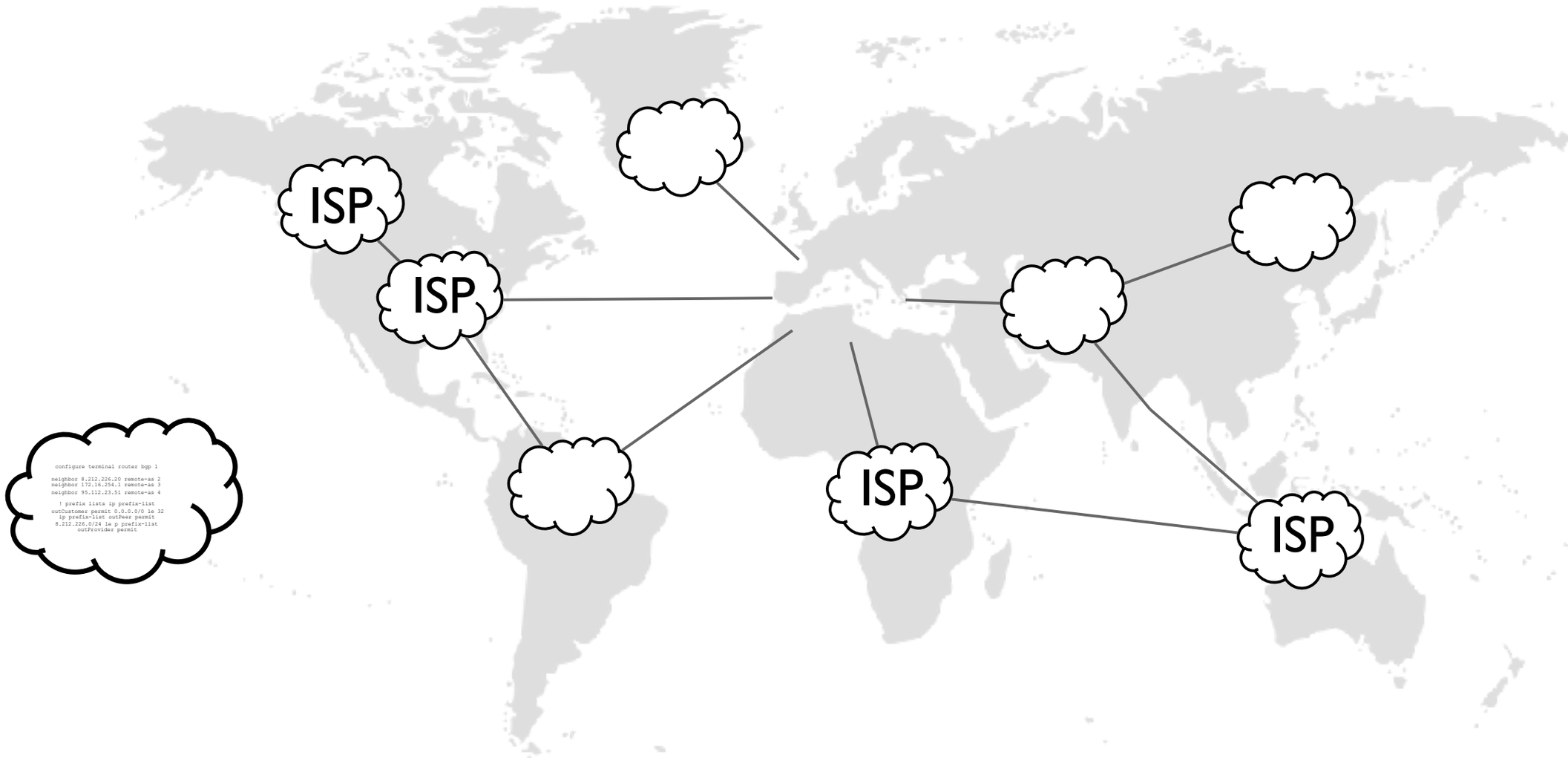
```
configure terminal router bgp 1

neighbor 8.212.226.20 remote-as 2
neighbor 172.16.254.1 remote-as 3
neighbor 95.112.23.51 remote-as 4

! prefix lists ip prefix-list
outCustomer permit 0.0.0.0/0 le 32
ip prefix-list outPeer permit
8.212.226.0/24 le p prefix-list
outProvider permit
```

IS

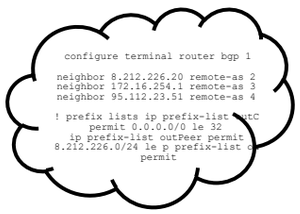
ISP



The Border Gateway Protocol



Specification



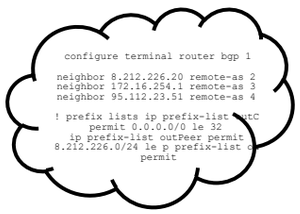
Configuration

Bagpipe

Spec Holds

Spec Violation

Specification



Configuration

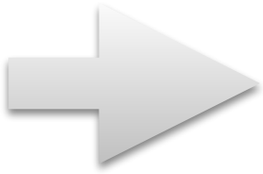
Bagpipe

Spec Holds

Spec Violation

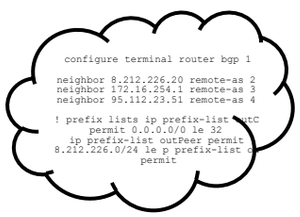
Bagpipe

Specification

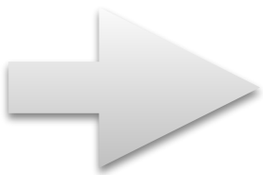


∀t:trace(

), check( , t)



Configuration

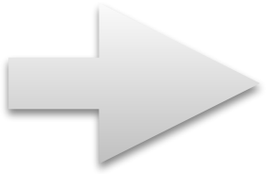


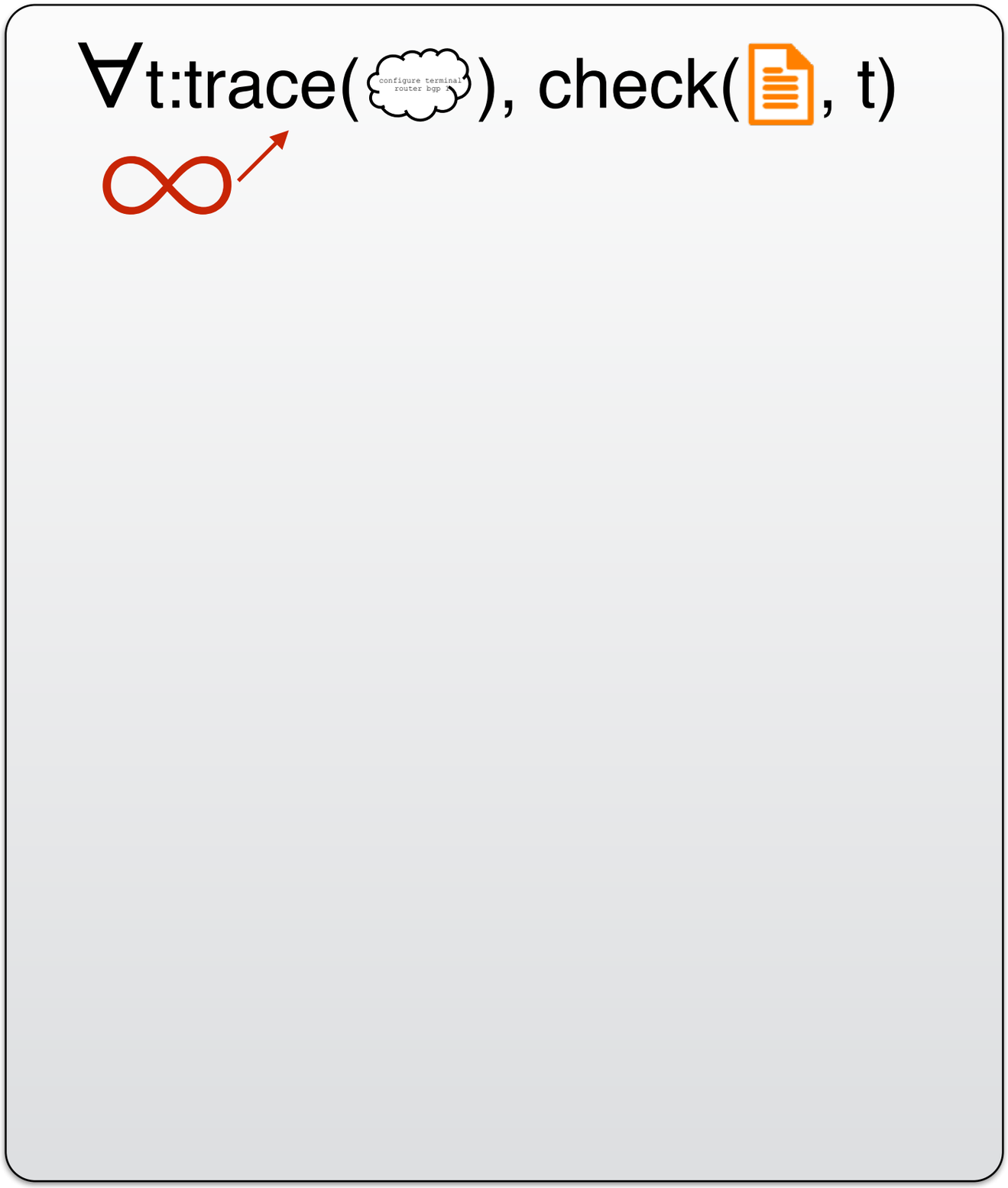
Spec Holds

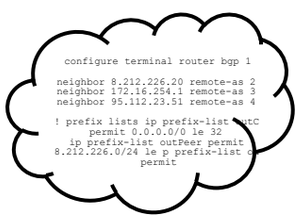
Spec Violation

Bagpipe

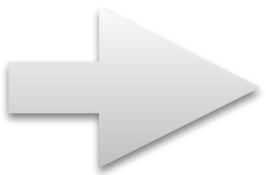
Specification

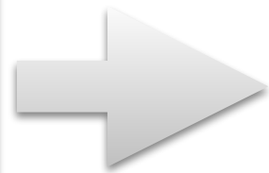






Configuration





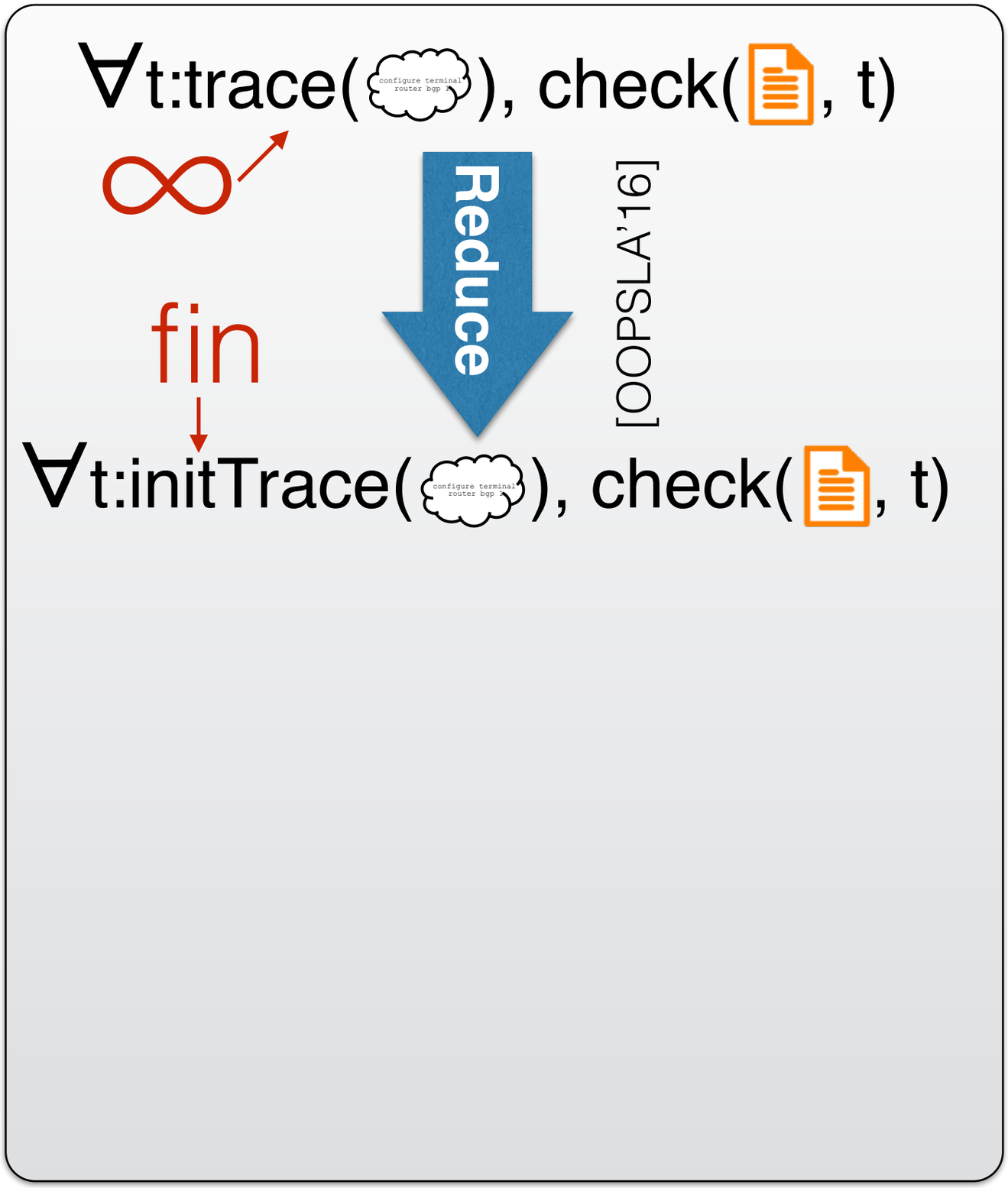
Spec Holds

Spec Violation

Bagpipe

Specification

Configuration



Spec Holds

Spec Violation

Bagpipe

Specification

Configuration

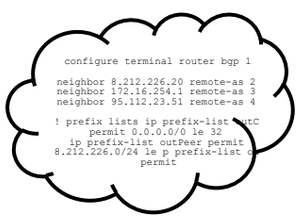
$\forall$t:trace(), check( , t)

fin

Reduce

[OOPSLA'16]

$\forall$t:initTrace(), check( , t)



Spec Holds

Spec Violation

Bagpipe

Specification

Configuration

$\forall$t:trace(configure terminal router bgp 1), check(, t)

∞

fin

$\forall$t:initTrace(configure terminal router bgp 1), check(, t)

Reduce

[OOPSLA'16]

Reduce ?

SMT

Spec Holds

Spec Violation

$\forall$t:initTrace(

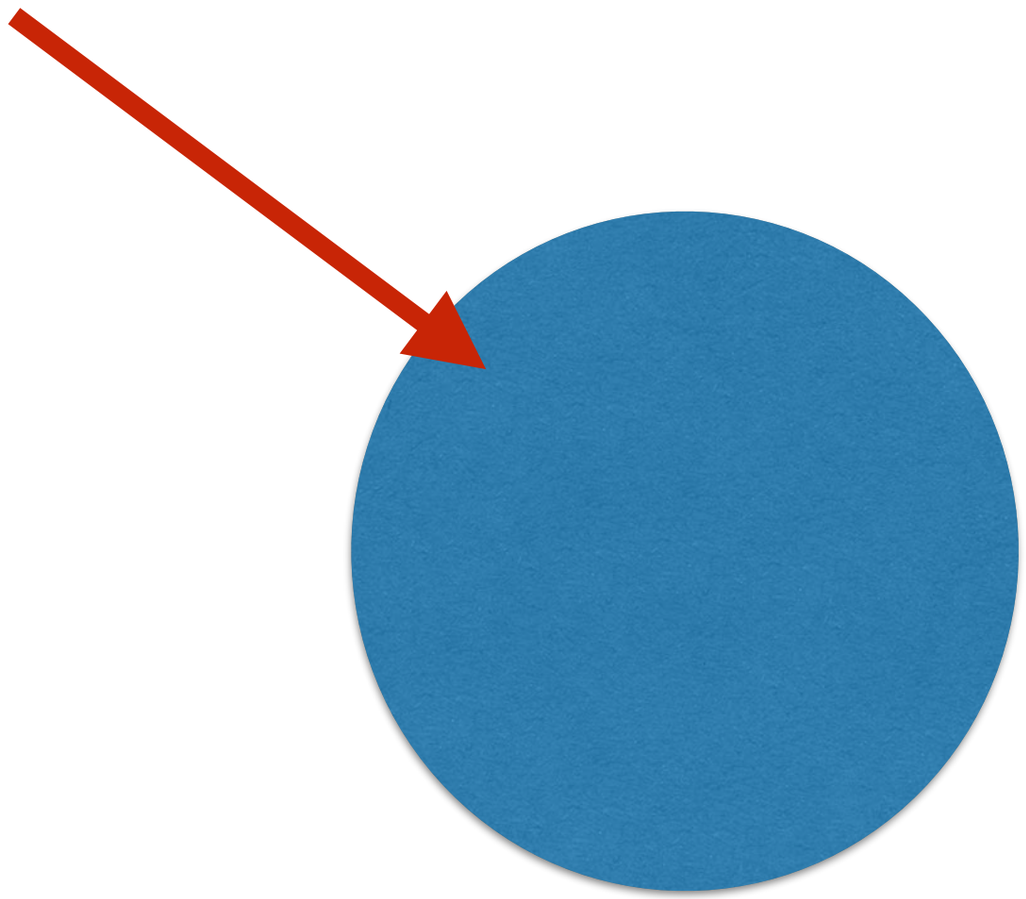
), check( , t)

$\forall$t:initTrace(

), check( , t)

initTrace(

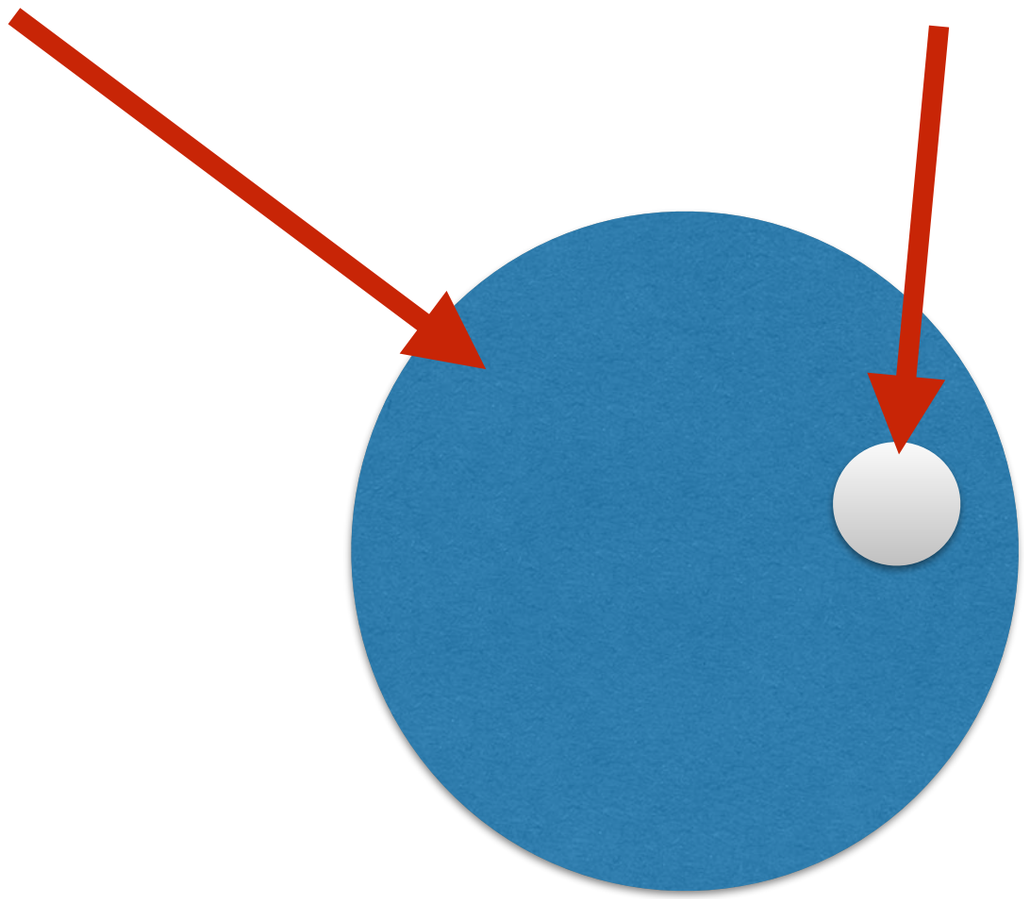
)



$$\forall t{:}initTrace(\text{configure terminal / router bgp 1 / neighbor 8.212.226.0}),\ check(\text{📄}, t)$$

$$initTrace(\text{configure terminal / router bgp 1})$$

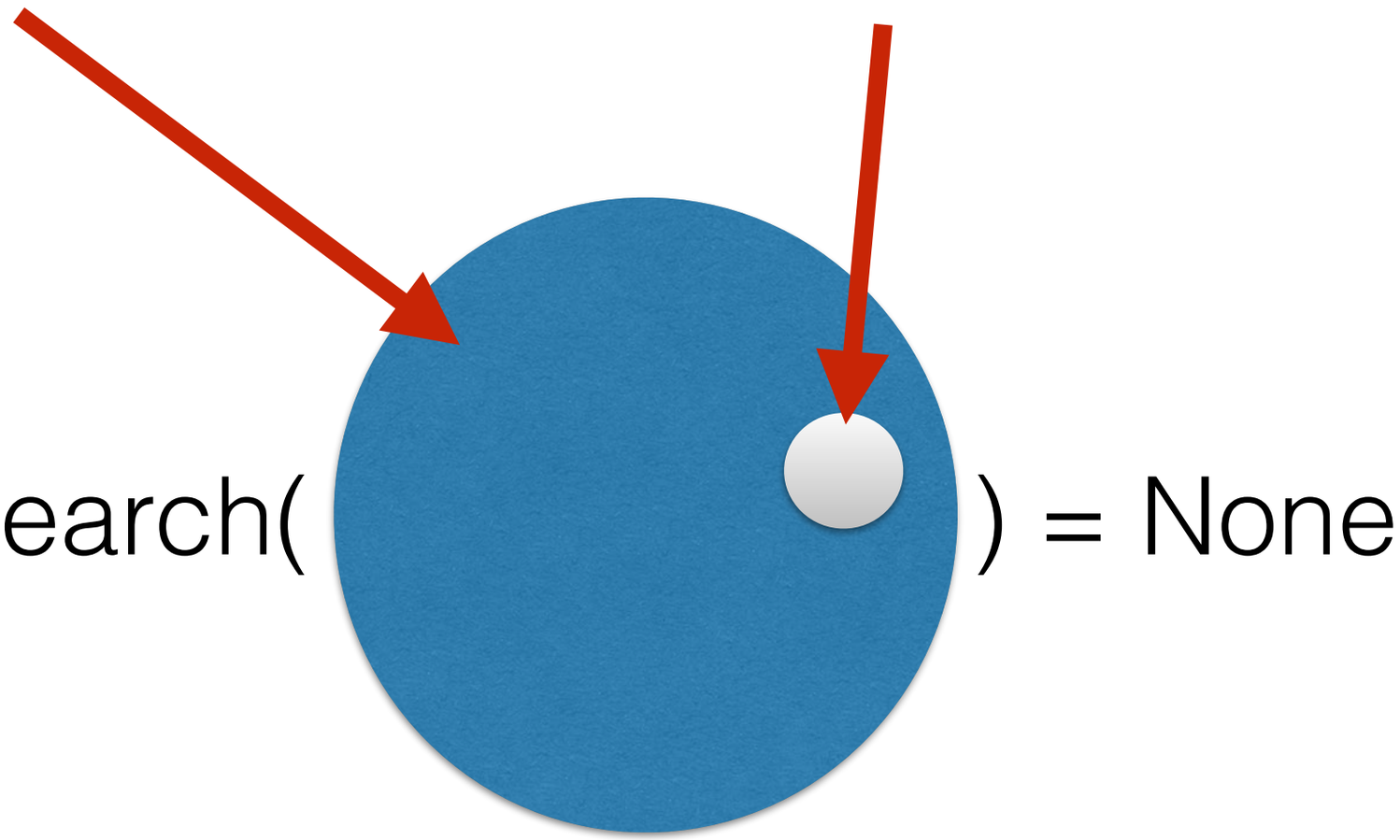
$$\{\ t{:}initTrace(\text{configure terminal / router bgp 1})\ |\ \neg check(\text{📄}, t)\ \}$$



$$\forall t{:}initTrace(\text{configure terminal / router bgp 1 / neighbor 8.212.226.0}),\ check(\text{📄}, t)$$

initTrace(☁ configure terminal / router bgp 1)

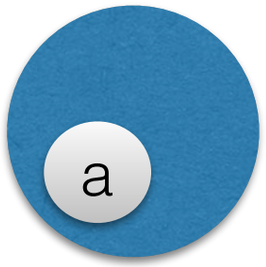
{ t:initTrace(☁ configure terminal / router bgp 1) | ¬check(📄, t) }

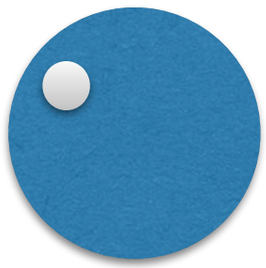


search( ) = None

SpaceSearch Interface

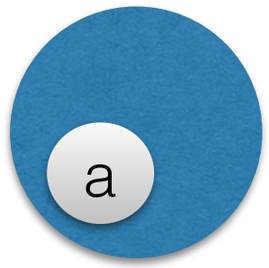
empty = 

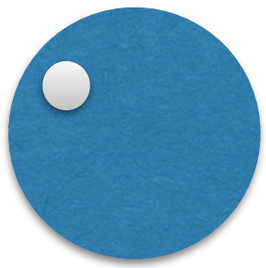
singleton(a) = 

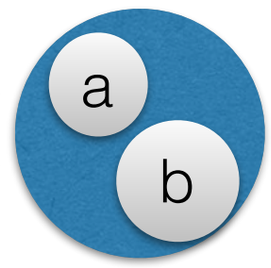
union( , ) = 

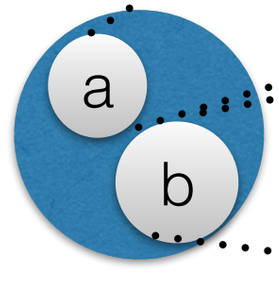
SpaceSearch Interface

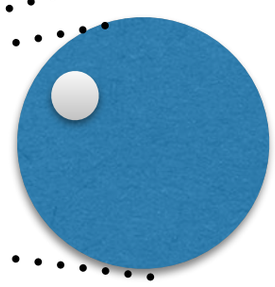
empty = 

singleton(a) = 

union( , ) = 

bind(S,f) = $U_{x:S}$ f(x) = bind(

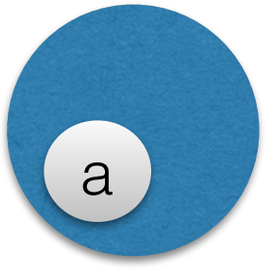
, 

f(a) f(b)  ) = 

SpaceSearch Interface

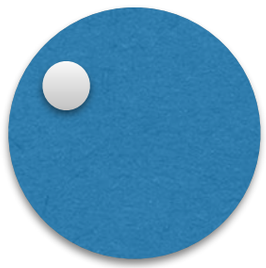
empty = 

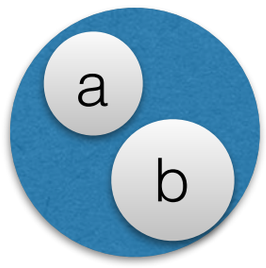
search() = None

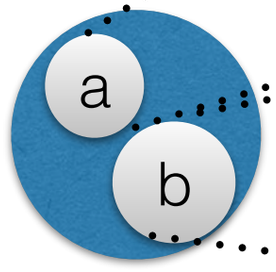
singleton(a) = 

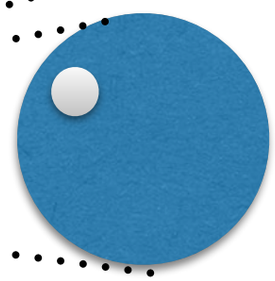
search(

) = Some a

union( , ) = 

bind(S,f) = $U_{x:S}$ f(x) = bind(

, 

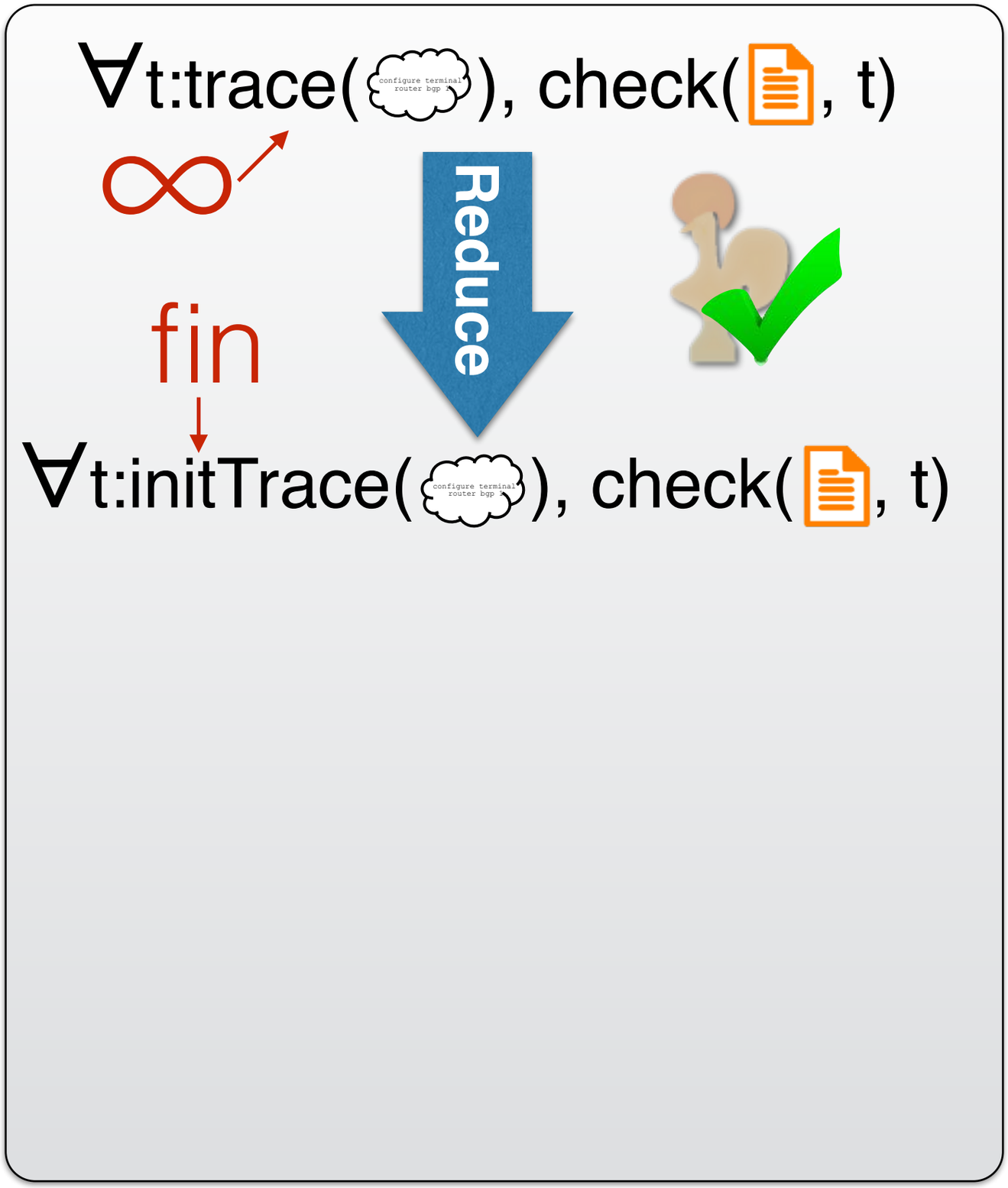


) = 

Bagpipe

Specification

Configuration



Spec Holds

Spec Violation

Bagpipe

Specification

Configuration

∀t:trace(), check( , t)

fin

Reduce

∀t:initTrace(), check( , t)

Reduce

```
search(
    bind(initTraceSpace( ), (λ t.
        if check( , t)
        then empty
        else singleton(t))) = None
```

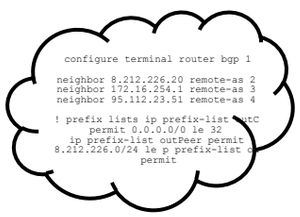
Spec Holds

Spec Violation

Bagpipe

Specification

Configuration



∀t:trace(), check( , t)

fin

Reduce

∀t:initTrace(), check( , t)

Reduce

search(
  bind(initTraceSpace(), (**λ** t.
    if check( , t)
    then empty
    else singleton(t))) = None

Extract ?

SMT

Spec Holds

Spec Violation

Meet ROSETTE

Meet ROSETTE

$\forall\ x\ y.\ (x \wedge y) \iff \neg(\neg x \vee \neg y)$ De Morgan's Law

Meet ROSETTE

$\forall\ x\ y.\ (x \wedge y) \iff \neg(\neg x \vee \neg y)$ De Morgan's Law

```
(let ((x (symbolic-bool))
      (y (symbolic-bool)))
```

Meet ROSETTE

$\forall\ x\ y.\ (x \wedge y) \iff \neg(\neg x \vee \neg y)$ De Morgan's Law

```
(let ((x (symbolic-bool))
      (y (symbolic-bool)))
     (eq? (and x y)
          (not (or (not x) (not y)))))
```

Meet ROSETTE

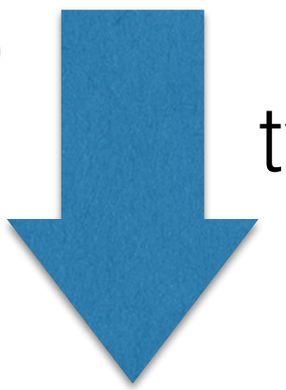
$\forall\ x\ y.\ (x \wedge y) \iff \neg(\neg x \vee \neg y)$ De Morgan's Law

```
(solve
 (let ((x (symbolic-bool))
       (y (symbolic-bool)))
  (if (eq? (and x y)
           (not (or (not x) (not y))))
      (assert false)
      'counter-example))
```

Meet 

$\forall\ x\ y.\ (x \wedge y) \iff \neg(\neg x \vee \neg y)$ De Morgan's Law

```
(solve
 (let ((x (symbolic-bool))
       (y (symbolic-bool)))
  (if (eq? (and x y)
           (not (or (not x) (not y))))
      (assert false)
      'counter-example))
```



type-driven state merging

```
(declare-const x Bool)
(declare-const y Bool)
(define-const a Bool (and x y))
(define-const b Bool (not (or (not x) (not y))))
(assert (not (and (=> a b) (=> b a))))
(check-sat)
```

SpaceSearch Extraction

=>

```
union(s,t) => (lambda (v) (if (symbolic-bool)
                              (s v) (t v)))

empty      => (lambda (v) (assert false))

search(s)  => (solve s)

single(a)  => (lambda (v) a)

bind(s,f)  => (lambda (v) (f (s v) v))
```

Bagpipe

Specification

Configuration

$\forall$t:trace(), check( , t)

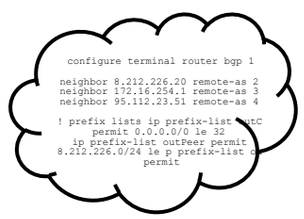
fin

Reduce

$\forall$t:initTrace(), check( , t)

Reduce

```
search(
    bind(initTraceSpace( ), (λ t.
        if check( , t)
        then empty
        else singleton(t))) = None
```



Spec Holds

Spec Violation

Bagpipe

Specification

Configuration

∀t:trace(), check( , t)

fin

Reduce

∀t:initTrace(), check( , t)

Reduce

search(
bind(initTraceSpace(), (λ
if check( , t)
then empty
else singleton(t))) = None

Extract

ROSETTE/SMT

Spec Holds

Spec Violation

Summary

SpaceSearch

- Interface & Semantics
- Extraction

Summary

SpaceSearch

- Interface & Semantics
- Extraction

More in the paper:

- Infinite Search Spaces
- Other Backends
- Parallelization
- Incrementalization

Evaluation

1. BGP Verification

Evaluation

1. BGP Verification

2. SQL Rewrite

Evaluation

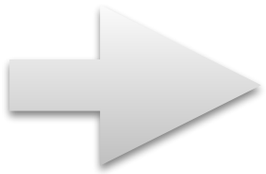
1. BGP Verification

2. SQL Rewrite

3. x86 Semantics

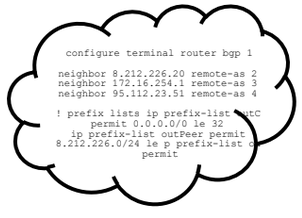
1. BGP Verification

Specification



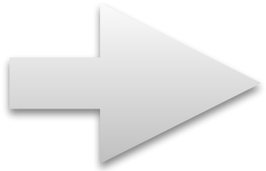
Correct

Counter Example

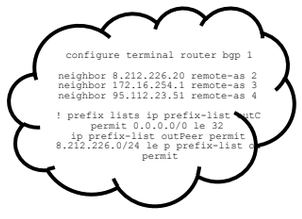


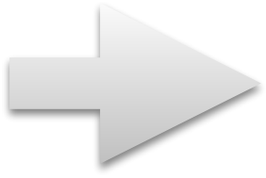
Configuration

1. BGP Verification



- 10 Juniper Scenarios
- No Martian
- Block To External
- Gao & Rexford





- Internet2 >100K
- BelWü >200K
- Selfnet >50

14 Specs Hold

19 Spec Violations, No False Positives

2. SQL Rewrite Verification

SELECT DISTINCT x.name
FROM Employee **AS** x, Employee **AS** y
WHERE x.name = y.name

=

SELECT DISTINCT name
FROM Employee

2. SQL Rewrite Verification

SELECT DISTINCT x.name
FROM Employee **AS** x, Employee **AS** y
WHERE x.name = y.name

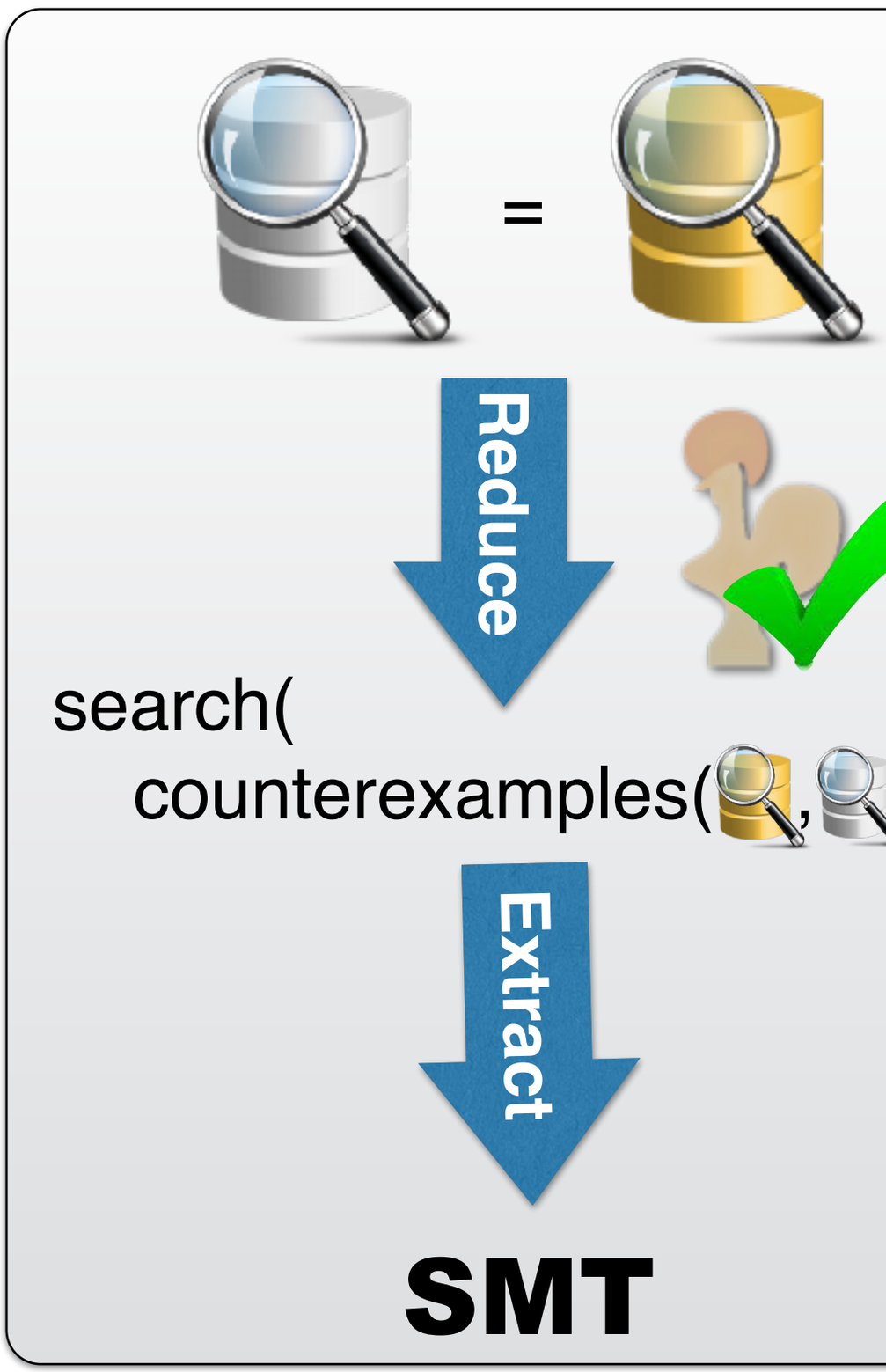
=

SELECT DISTINCT name
FROM Employee

ConjunctiveQuery:
SELECT DISTINCT a,b **FROM** A,B **WHERE** a = b, b = c

2. SQL Rewrite Verification

Query A



Query B

Equal

Counter Example

3. x86 Semantics Validation

```
search(
    bind(int32, (λ x.
    bind(int32, (λ y.
        if stoke(ADD x, y) =
           rocksalt(ADD x, y)
        then empty
        else singleton(x,y)))))
```

Extract

SMT

ADD x, y

Instruction

Equal

Counter Example

3. x86 Semantics Validation

```
search(
    bind(int32, (λ x.
    bind(int32, (λ y.
        if stoke(ADD x, y) =
           rocksalt(ADD x, y)
        then empty
        else singleton(x,y)))))
```

ADD x, y

Instruction

Extract

SMT

Equal

Counter Example

Bugs found:

- 7 Rocksalt Bugs
- 1 Stoke Bug

Related Work

- Solver Aided Languages:
  Rosette Torlak et al. PLDI'14
  Smten Uhler et al. CAV'13

- Solver Aided Tool Verification:
  XCert Tatlock et al. PLDI'10

- Verified SAT Solvers & SAT Tactics:
  Marić TCS'10
  Oe et al. VMCAI'12

Thank You

github.com/konne88/SpaceSearch

Konstantin Weitz Steven S. Lyubomirsky Stefan Heule Emina Torlak Michael D. Ernst Zachary Tatlock