

# The bane of network operations: Networks are evolved, not designed

Ratul Mahajan

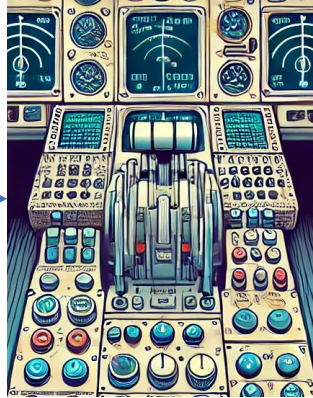


PAUL G. ALLEN SCHOOL  
OF COMPUTER SCIENCE & ENGINEERING

# The unfortunate lifecycle of a network



Starts out simple



Complexity grows

Hard to operate



Abstractions harness complexity



Abstractions are "punctured"

Even harder to operate!

# Example (1/3): Configuration synthesis



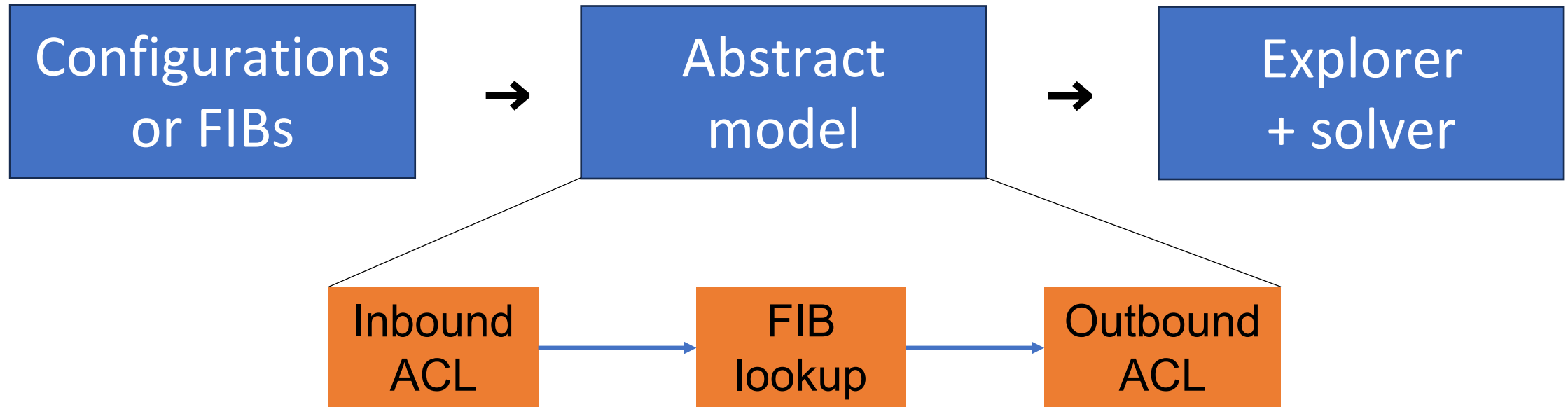
What if the topology or policy evolves?

What if a new type of device is introduced?

How do we work around a buggy device behavior?

Is the synthesis process stable?

# Example (2/3): Network verification

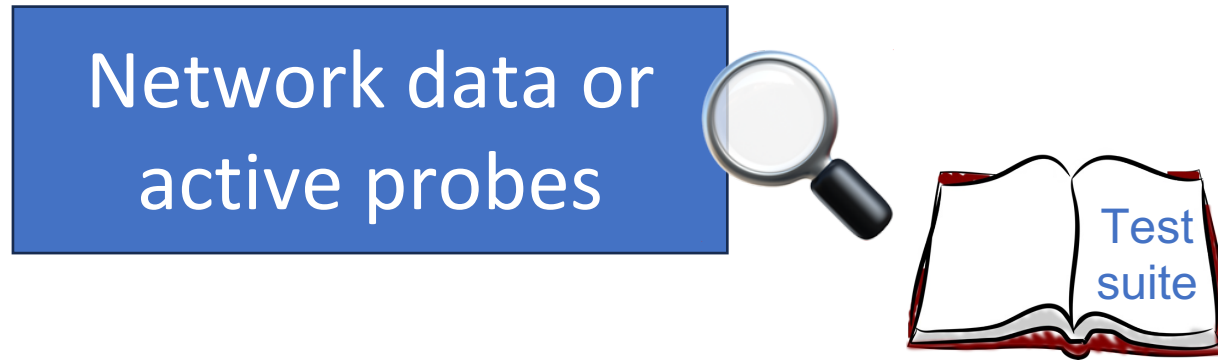


When does NAT happen?

When does firewalling happen?

Can ACLs and firewall rules refer to pre-NAT fields?

# Example (3/3): Test suites



Which tests are relevant as the network evolves?

Which tests are not redundant?

Are new behaviors being sufficiently tested?

# Downsides of “static” tools

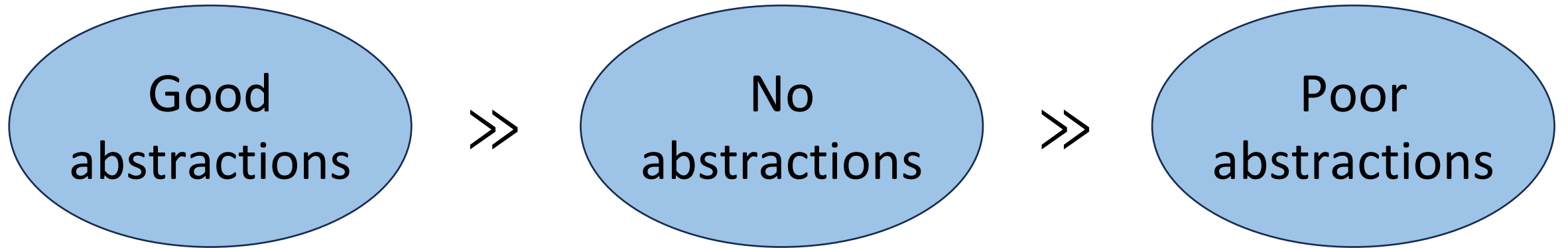
Can leave the network worse off in the long term

→ Hurt agility with questionable manageability benefit

Engineers with foresight won't adopt them

Conjecture: This is why declarative tools lose to imperative tools

# Conjecture



# Building evolution-friendly tools

Plan for evolution

Use more flexible, lower-level abstractions

Expose and quantify drift



# Evolution-friendly configuration synthesis

## Don't Mind the Gap: Bridging Network-wide Objectives and Device-level Configurations

Ryan Beckett  
Princeton

Ratul Mahajan  
Microsoft

Todd Millstein  
UCLA

Jitendra Padhye  
Microsoft

David Walker  
Princeton

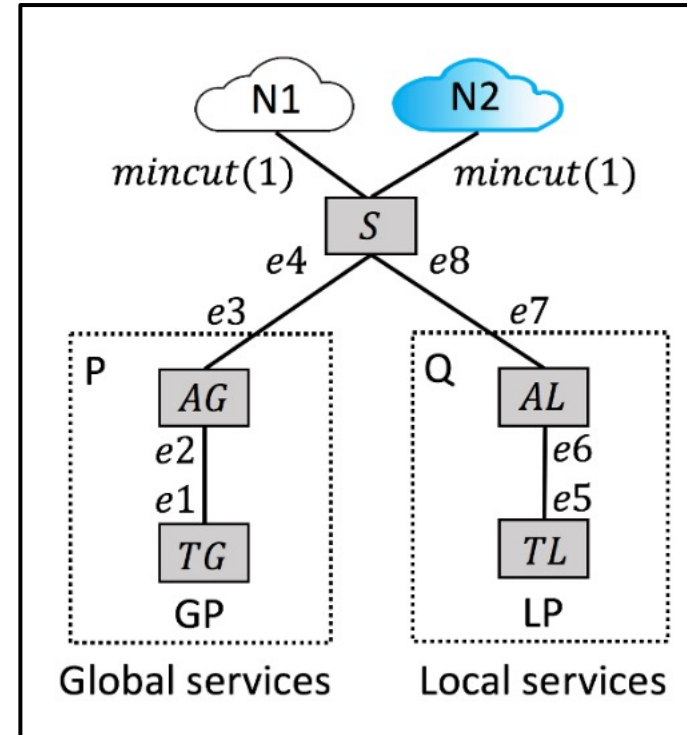
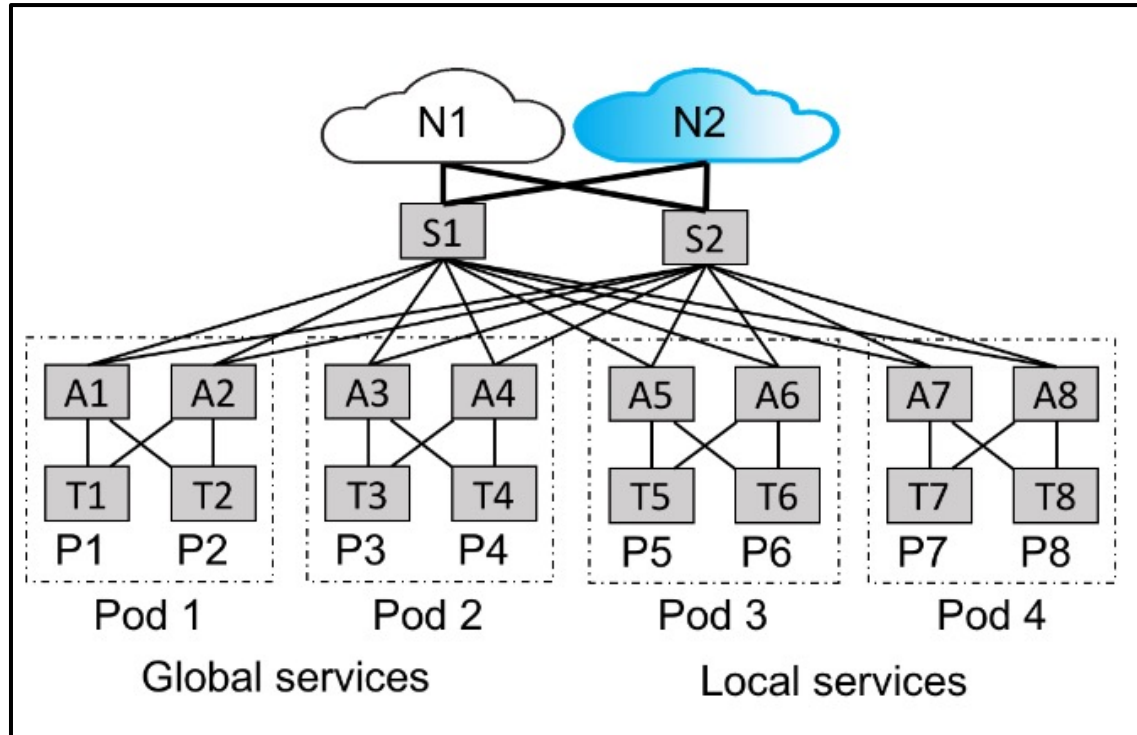
[SIGCOMM 2016]

## Network Configuration Synthesis with Abstract Topologies

Ryan Beckett\*   Ratul Mahajan<sup>†‡</sup>   Todd Millstein<sup>†§</sup>   Jitendra Padhye<sup>‡</sup>   David Walker\*  
Princeton University, USA\*   Intentionet, USA<sup>†</sup>   Microsoft, USA<sup>‡</sup>   UCLA, USA<sup>§</sup>

[PLDI 2017]

# Propane/AT (Abstract Topologies)



- $e1 = AG$
- $e2 = TG$
- $e5 = AL$
- $e6 = TL$
- $AG = AL$
- $AG \leq S$
- $e3 \geq 2$
- $e4 \geq 1$
- $e7 \geq 2$
- $S \bmod AG = 0$
- $S \bmod AL = 0$
- $2 \leq S \leq 4$

# Evolution-friendly network modeling

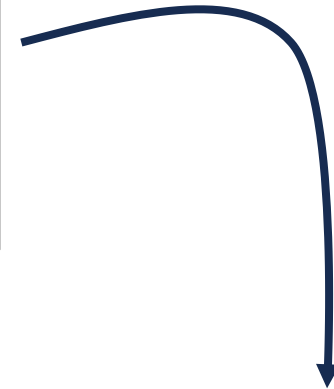
## **A General Approach to Network Configuration Analysis**

*Ari Fogel Stanley Fung Luis Pedrosa Meg Walraed-Sullivan*

*Ramesh Govindan Ratul Mahajan Todd Millstein*

University of California, Los Angeles University of Southern California Microsoft Research

[NSDI 2015]



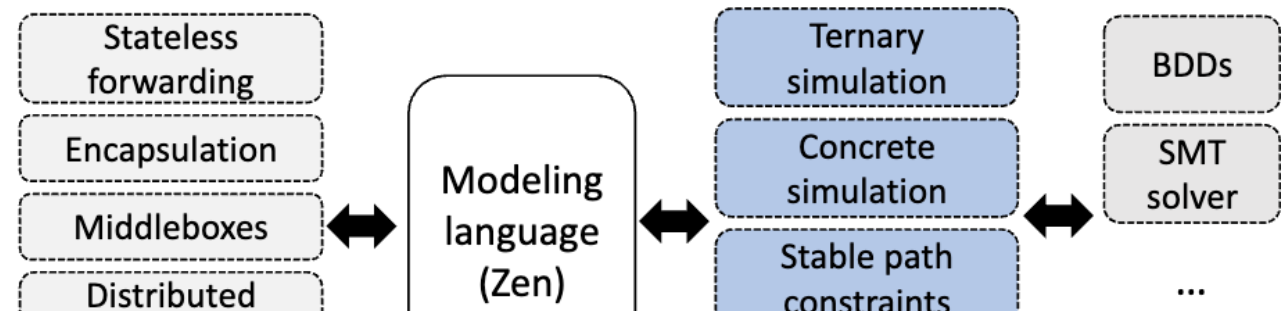
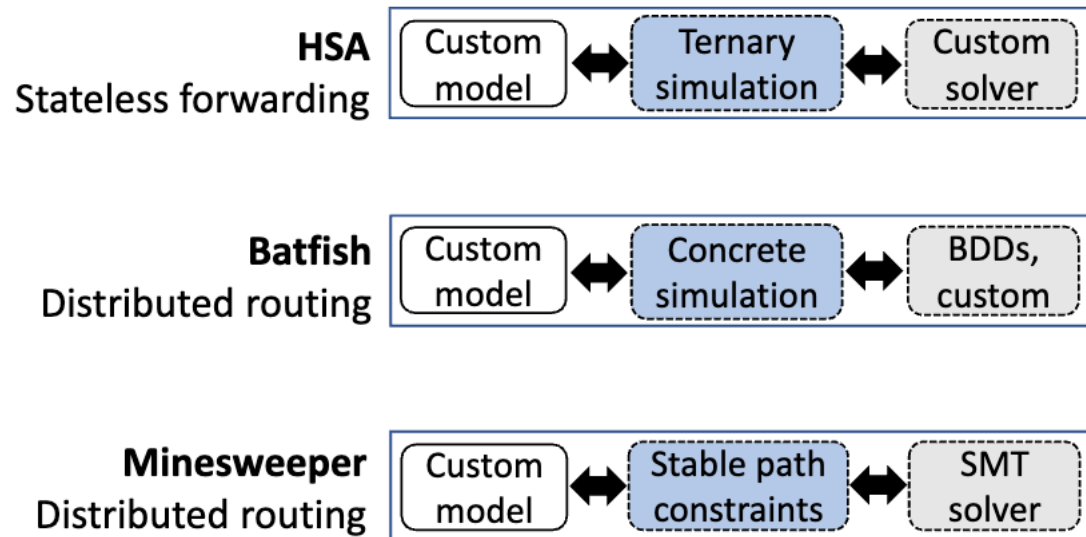
## **A General Framework for Compositional Network Modeling**

Ryan Beckett  
Microsoft Research

Ratul Mahajan  
University of Washington, Intentionet

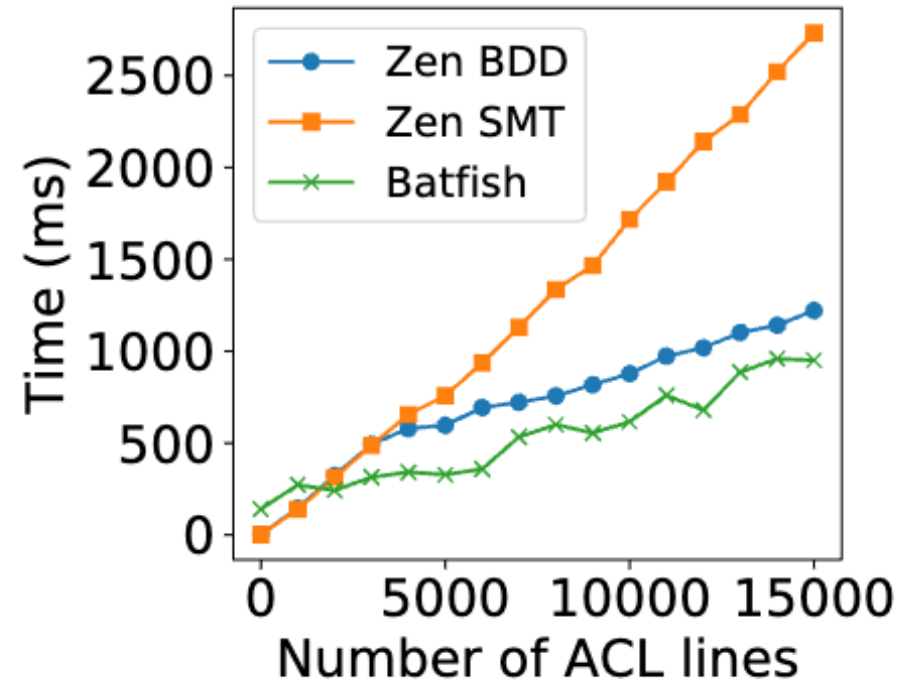
[HotNets 2020]

# Modeling networks using Zen



```
Zen<Option<Packet>> Fwd(Intf[] path, Zen<Packet> p) {  
    Zen<Option<Packet>> x = Some(p);  
    for (int i = 0; i < path.Length; i++) {  
        var intfIn = path[i];  
        var intfOut = path[i + 1];  
        x = If(x.HasValue, FwdIn(intfIn, x.Value), x);  
        x = If(x.HasValue, FwdOut(intfOut, x.Value), x);  
    }  
}
```

# Performance of Zen vs Batfish



# Evolution-friendly network verification



## **Relational Network Verification**

Xieyang Xu<sup>[w]</sup> Yifei Yuan<sup>[a]</sup> Zachary Kincaid<sup>[p]</sup> Arvind Krishnamurthy<sup>[w]</sup>  
Ratul Mahajan<sup>[w]</sup> David Walker<sup>[p]</sup> Ennan Zhai<sup>[a]</sup>

<sup>[w]</sup>University of Washington    <sup>[a]</sup>Alibaba Cloud    <sup>[p]</sup>Princeton University

**[SIGCOMM 2024]**

# Evolution-friendly test suites

## Test Coverage Metrics for the Network

Xieyang Xu  
University of Washington

Ryan Beckett  
Microsoft

Karthick Jayaraman  
Microsoft

Ratul Mahajan  
University of Washington, Intentionet

David Walker  
Princeton University

[SIGCOMM 2021]

## Test Coverage for Network Configurations

Xieyang Xu<sup>1</sup> Weixin Deng<sup>1</sup> Ryan Beckett<sup>2</sup> Ratul Mahajan<sup>1</sup> David Walker<sup>3</sup>

<sup>1</sup>*University of Washington*   <sup>2</sup>*Microsoft*   <sup>3</sup>*Princeton University*

[NSDI 2023]

# NetCov: Coverage for network configurations

Current view: [top level - co](#)  
 Test: [internet2.initi](#)  
 Date: [2022-09-20 14](#)

Filename
<a href="#">atla.conf</a>
<a href="#">chic.conf</a>
<a href="#">cleve.conf</a>
<a href="#">hous.conf</a>
<a href="#">kans.conf</a>
<a href="#">losa.conf</a>
<a href="#">newy32aoa.conf</a>
<a href="#">salt.conf</a>
<a href="#">seat.conf</a>
<a href="#">wash.conf</a>

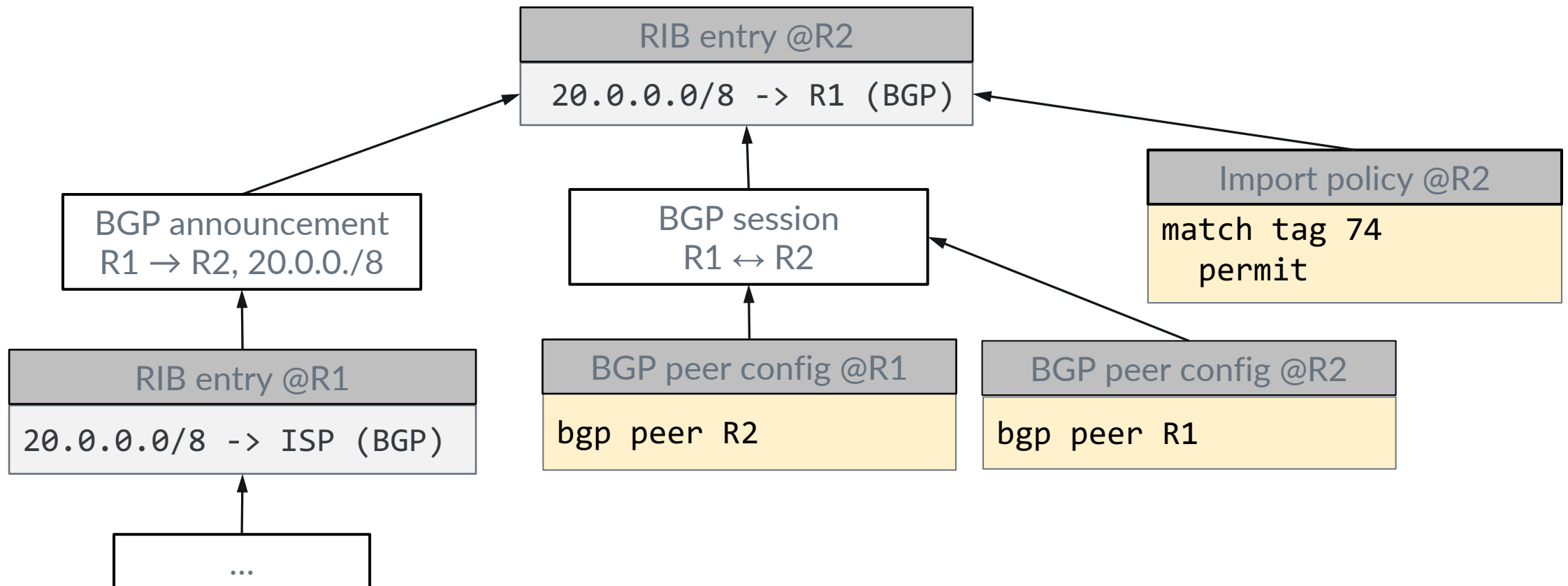
```

12105 /* reject routes we should never accept */
12106 policy-statement SANITY-IN {
12107 /* Reject any BGP prefix if a private AS is in the path */
12108 term block-private-asn {
12109     from as-path PRIVATE;
12110     then reject;
12111 }
12112 /* Reject any BGP NLRI=Unicast prefix if a commercial ISP's AS is in the path */
12113 term block-commercial-asn {
12114     from as-path COMMERCIAL;
12115     to rib inet.0;
12116     then reject;
12117 }
12118 term block-nlr-transit {
12119     from as-path NLR;
12120     then reject;
12121 }
12122 /* Reject BGP prefixes that should never appear in the routing table */
12123 term block-martians {
12124     from {
12125         /* default */
12126         route-filter 0.0.0.0/0 exact;
12127         /* rfc 1918 */
12128         route-filter 10.0.0.0/8 orlonger;
12129         /* rfc 3330 - loopback */
12130         route-filter 127.0.0.0/8 orlonger;
12131         /* rfc 3330 - link-local */
12132         route-filter 169.254.0.0/16 orlonger;
12133         /* rfc 1918 */
12134         route-filter 172.16.0.0/12 orlonger;
12135         /* iana reserved */
12136         route-filter 192.0.2.0/24 orlonger;
12137         /* 6to4 relay */
12138         route-filter 192.88.99.1/32 exact;
12139         /* rfc 1918 */
12140         route-filter 192.168.0.0/16 orlonger;
12141         /* rfc 2544 - network device benchmarking */
12142         route-filter 198.18.0.0/15 orlonger;
12143         /* rfc 3171 - multicast group addresses */
12144         route-filter 224.0.0.0/4 orlonger;
12145         /* rfc 3330 */
12146         route-filter 240.0.0.0/4 orlonger;
12147     }
12148     then reject;
12149 }
12150 /* Reject BGP prefixes which Abilene originates */
  
```

	Total	Coverage
2	64886	26.1 %
⬇		
	1211 / 5019	
	4376 / 10800	
	1156 / 3512	
	1196 / 4801	
	1235 / 6178	
	1832 / 8960	
	770 / 6545	
	568 / 3063	
	1845 / 6030	
	2723 / 9978	



# NetCov maps tested forwarding state to config lines



# Summary

Networks are evolved, not designed

Risk getting stuck with poor abstractions

Design networking tools for evolution

- Plan for evolution
- Use more flexible, lower-level abstractions
- Expose drift