Invisinets:
Removing Networking from Cloud Networks

Sarah McClure[1], Zeke Medley[1], Deepak Bansal[2], Karthick Jayaraman[2], Ashok Narayanan[3], Jitendra Padhye[2], Sylvia Ratnasamy[1,3], Anees Shaikh[3], Rishabh Tewari[2]

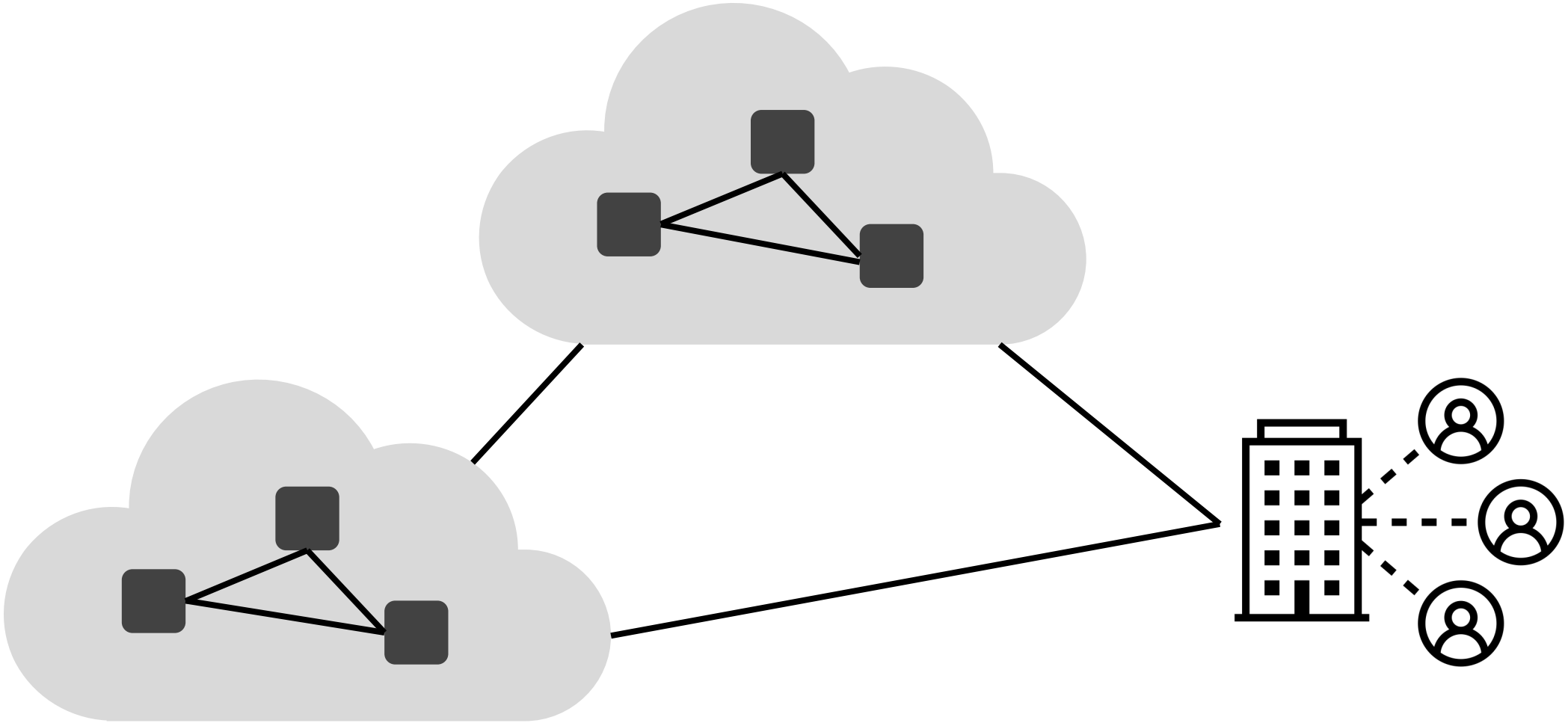
1


2

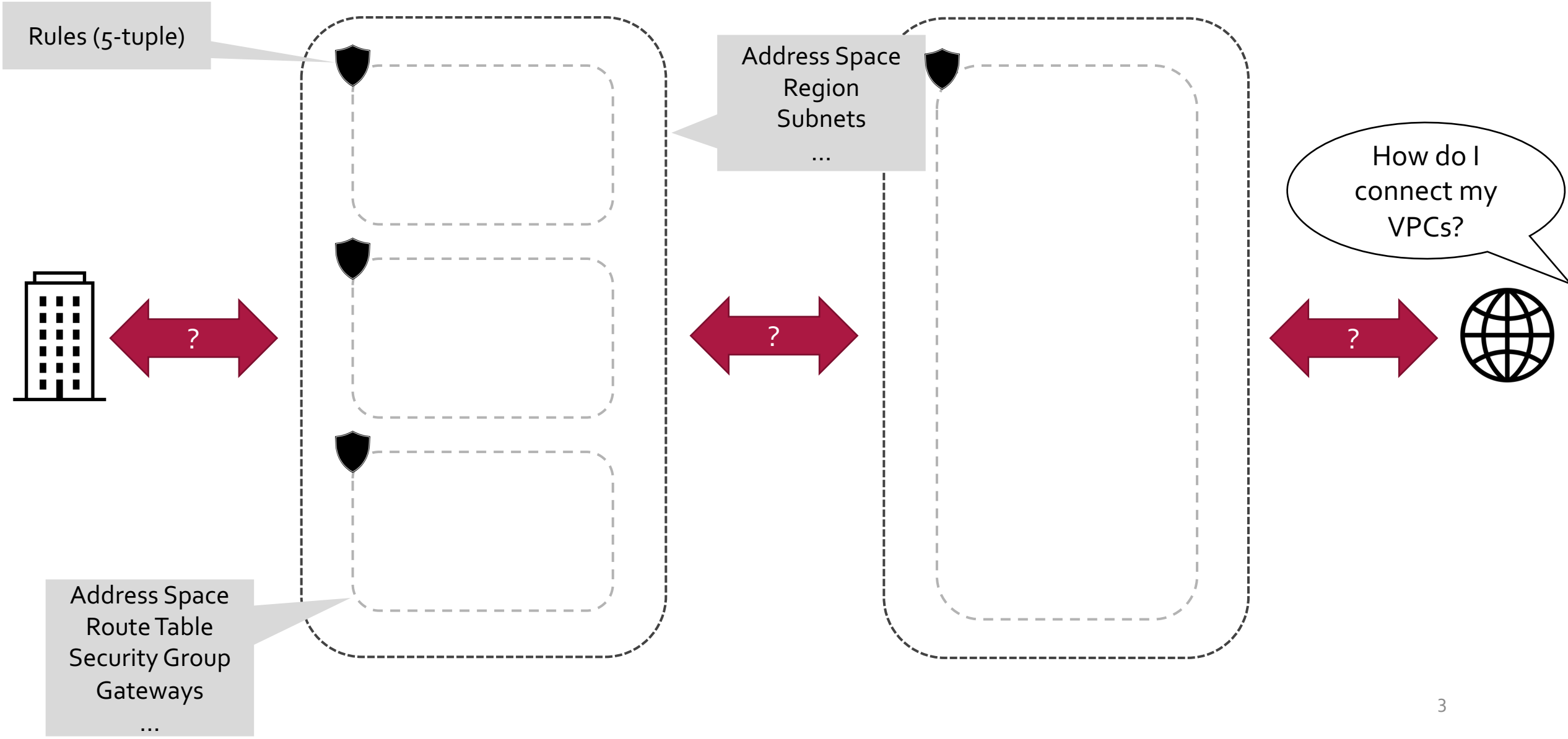

3

sarah@cs.berkeley.edu

Cloud IAAS Deployments



Walkthrough



Walkthrough

Key choices in AWS network design: VPC peering

AWS VPC to VPC Connection

AWS Connectivity Options – Choosing the Right Tool for the Job

January 7

VPN or Direct Connect? AWS Network Services Compared

How to connect AWS VPC? How to use peering? tutorial,

05/12/2

Connect Azure using VPN Gateway to

Docs › Integrations › Cloud servers

How to cre

Tutorial

UPDATED ON: JANUARY 6, 2023 •

Step by Step

Building a Scalable and Secure Multi-VPC AWS Network Infrastructure

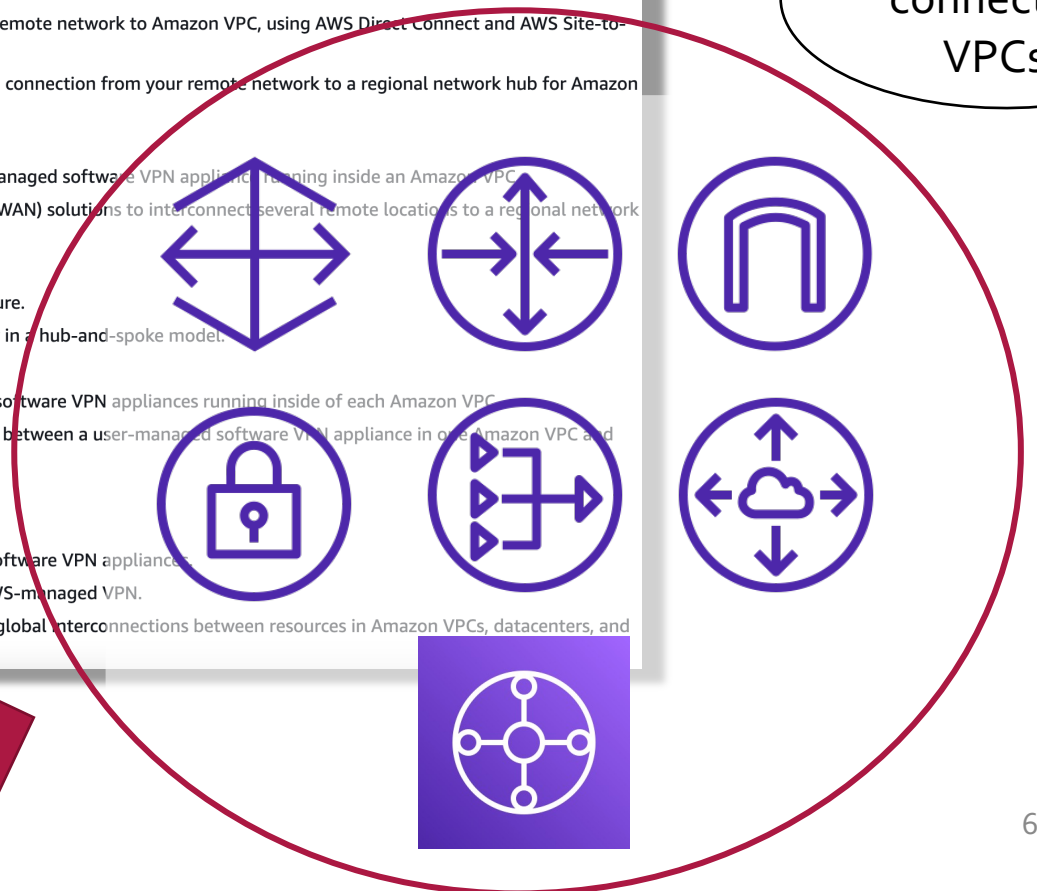
AWS Whitepaper

How do I connect my VPCs?

Walkthrough

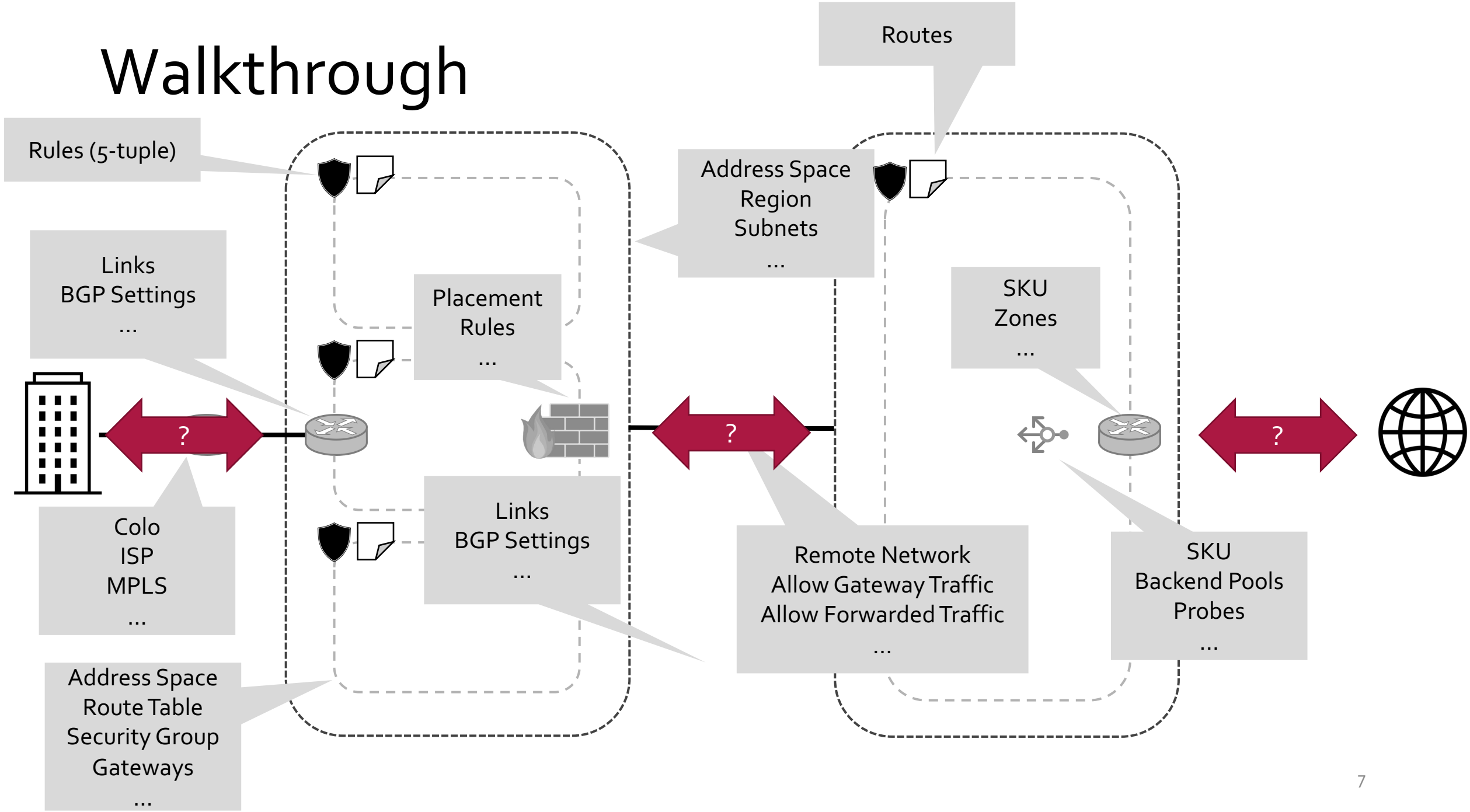
- Network-to-Amazon VPC connectivity options
  - AWS Site-to-Site VPN – Describes establishing a managed IPsec VPN connection from your network equipment on a remote network to Amazon VPC.
  - AWS Transit Gateway + AWS Site-to-Site VPN – Describes establishing a managed IPsec VPN connection from your network equipment on a remote network to a regional network hub for Amazon VPCs, using AWS Transit Gateway.
  - AWS Direct Connect - Describes establishing a private, logical connection from your remote network to Amazon VPC, using AWS Direct Connect.
  - AWS Direct Connect + AWS Transit Gateway – Describes establishing a private, logical connection from your remote network to a regional network hub for Amazon VPCs, using AWS Direct Connect and AWS Transit Gateway.
  - AWS Direct Connect + AWS Site-to-Site VPN – Describes establishing a private, encrypted connection from your remote network to Amazon VPC, using AWS Direct Connect and AWS Site-to-Site VPN.
  - AWS Direct Connect + AWS Transit Gateway + AWS Site-to-Site VPN – Describes establishing a private, encrypted connection from your remote network to a regional network hub for Amazon VPCs, using AWS Direct Connect and AWS Transit Gateway.
  - AWS VPN CloudHub – Describes establishing a hub-and-spoke model for connecting remote branch offices.
  - Software VPN – Describes establishing a VPN connection from your equipment on a remote network to a user-managed software VPN appliance running inside an Amazon VPC.
  - AWS Transit Gateway + SD-WAN solutions - Describes the integration of sofwaredefined-wide area network (SD-WAN) solutions to interconnect several remote locations to a regional network hub for Amazon VPCs, using the AWS backbone or the internet as a transit network.
- Amazon VPC-to-Amazon VPC connectivity options
  - VPC peering – Describes connecting Amazon VPCs within and across regions using the Amazon VPC peering feature.
  - AWS Transit Gateway – Describes connecting Amazon VPCs within and across regions using AWS Transit Gateway in a hub-and-spoke model.
  - AWS PrivateLink – Describes connecting Amazon VPCs with VPC interface endpoints and VPC endpoint services.
  - Software VPN – Describes connecting Amazon VPCs using VPN connections established between user-managed software VPN appliances running inside of each Amazon VPC.
  - Software VPN-to-AWS Site-to-Site VPN – Describes connecting Amazon VPCs with a VPN connection established between a user-managed software VPN appliance in one Amazon VPC and AWS Site-to-Site VPN attached to the other Amazon VPC.
- Software remote access-to-Amazon VPC connectivity options
  - AWS Client VPN – Describes connecting software remote access to Amazon VPC, leveraging AWS Client VPN.
  - Software client VPN – Describes connecting software remote access to Amazon VPC, leveraging user-managed software VPN appliances.
- Transit VPC - Describes establishing a global transit network on AWS using a software VPN in conjunction with an AWS-managed VPN.
- AWS Cloud WAN - Describes establishing a managed wide area network (WAN) to easily build, manage, and monitor global interconnections between resources in Amazon VPCs, datacenters, and remote branches.

How do I connect my VPCs?

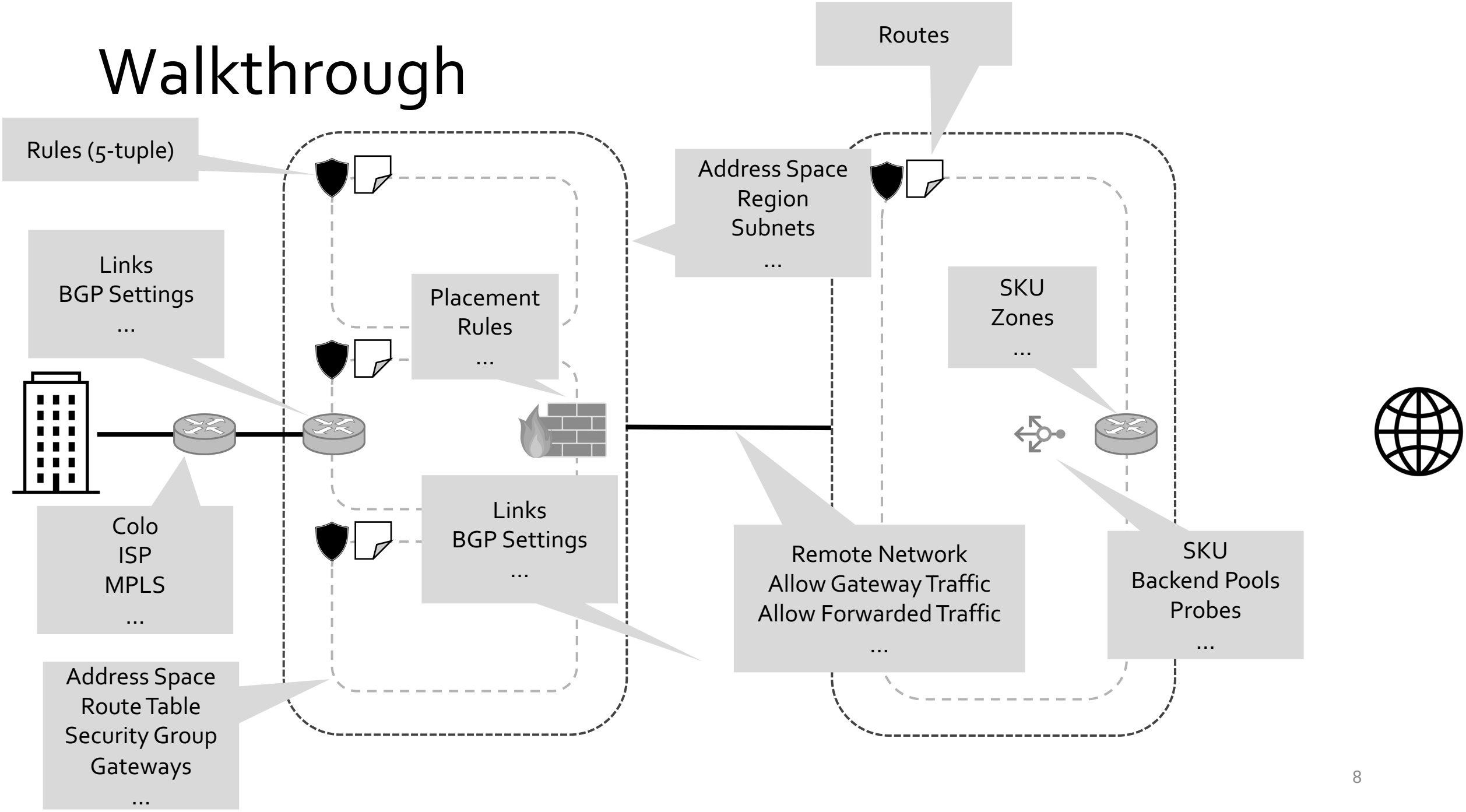


Ways out of your private address space

Walkthrough

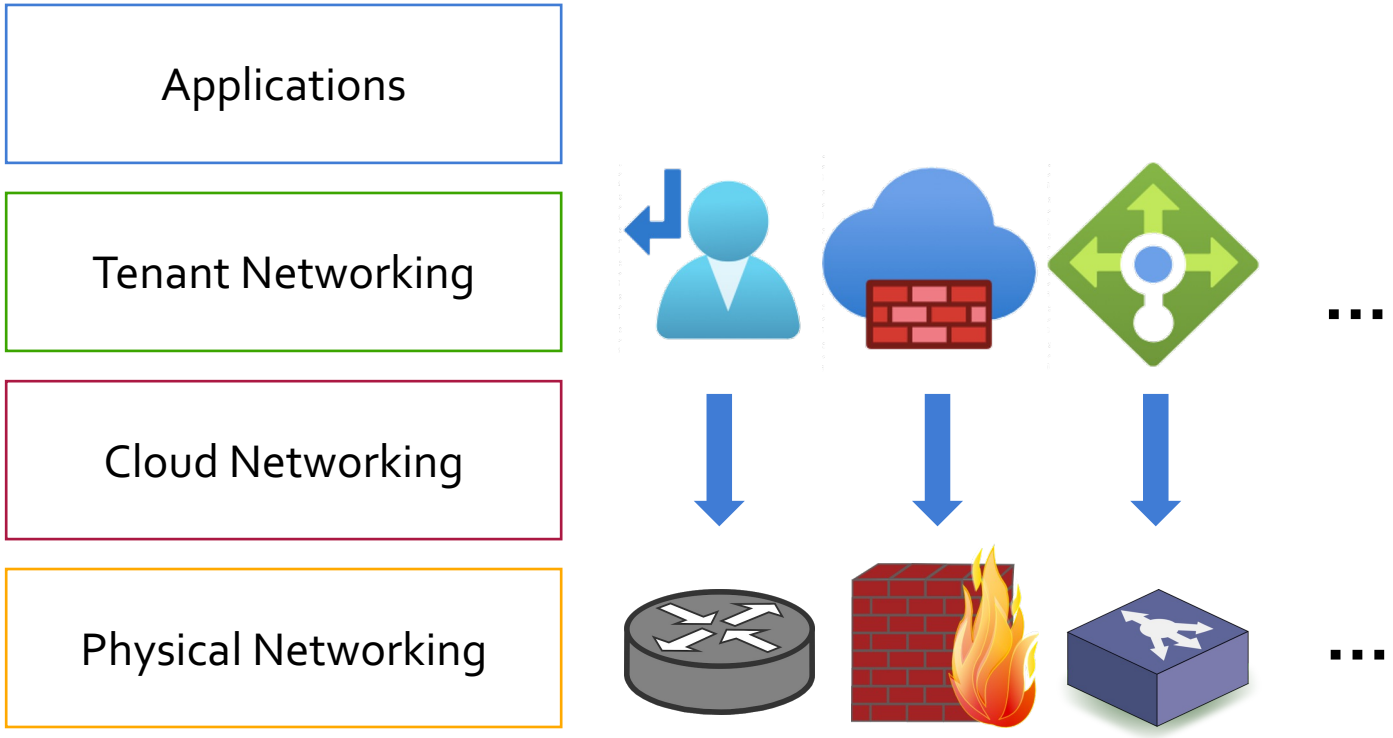


Walkthrough



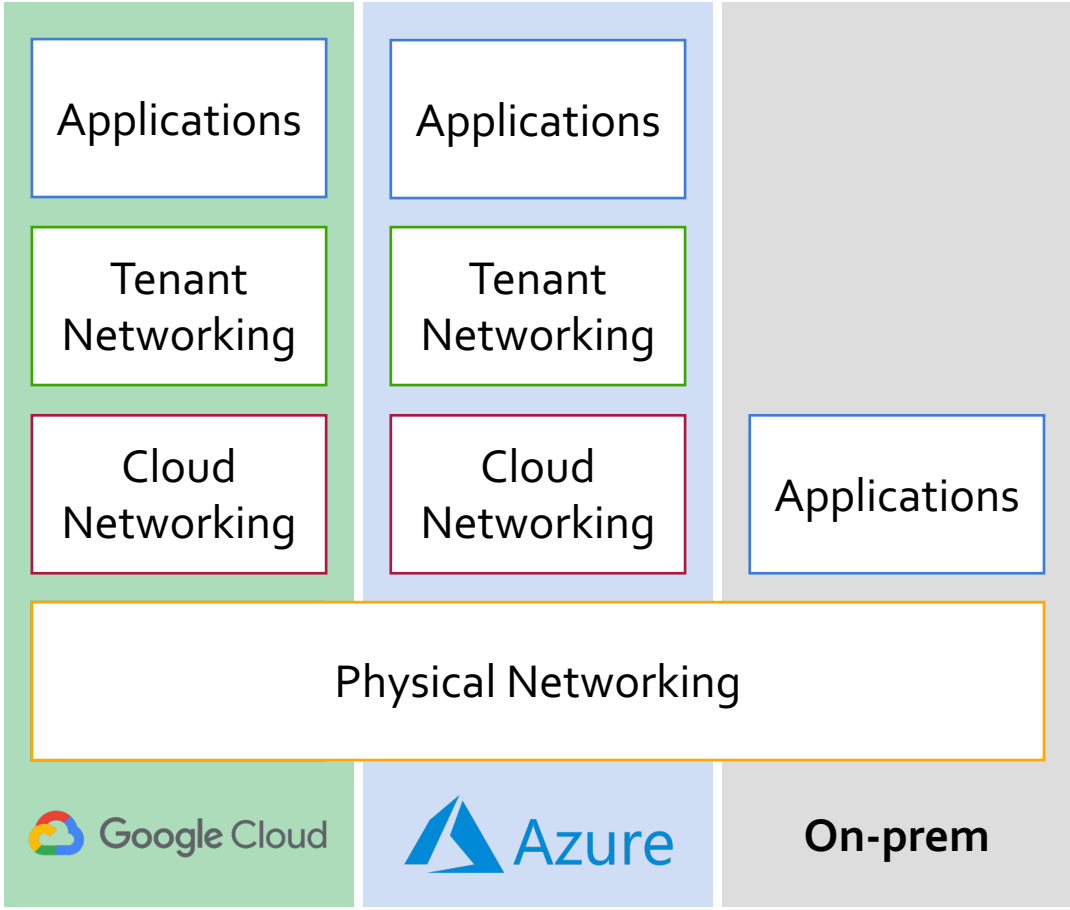
Problem

❌ **Low-level abstractions**
❌ **Private virtual networks →**
Required to construct connectivity



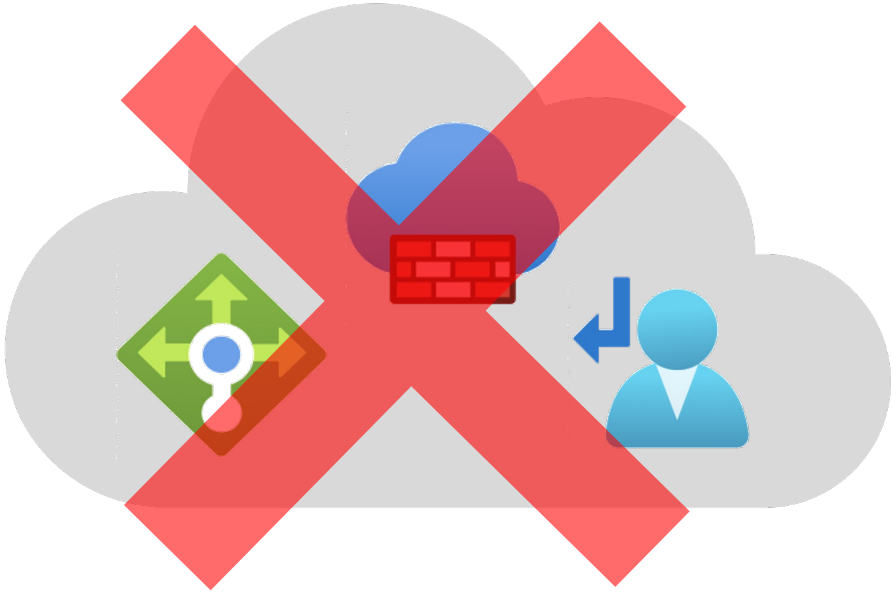
Problem

❌ **Low-level abstractions**
❌ **Private virtual networks →**
Required to construct connectivity
❌ **Fragmented across clouds**

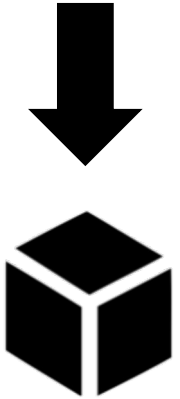


Approach

1. Make connectivity trivial
2. Design around tenant goals
3. Specify these goals on endpoints
4. Put the burden on the cloud provider



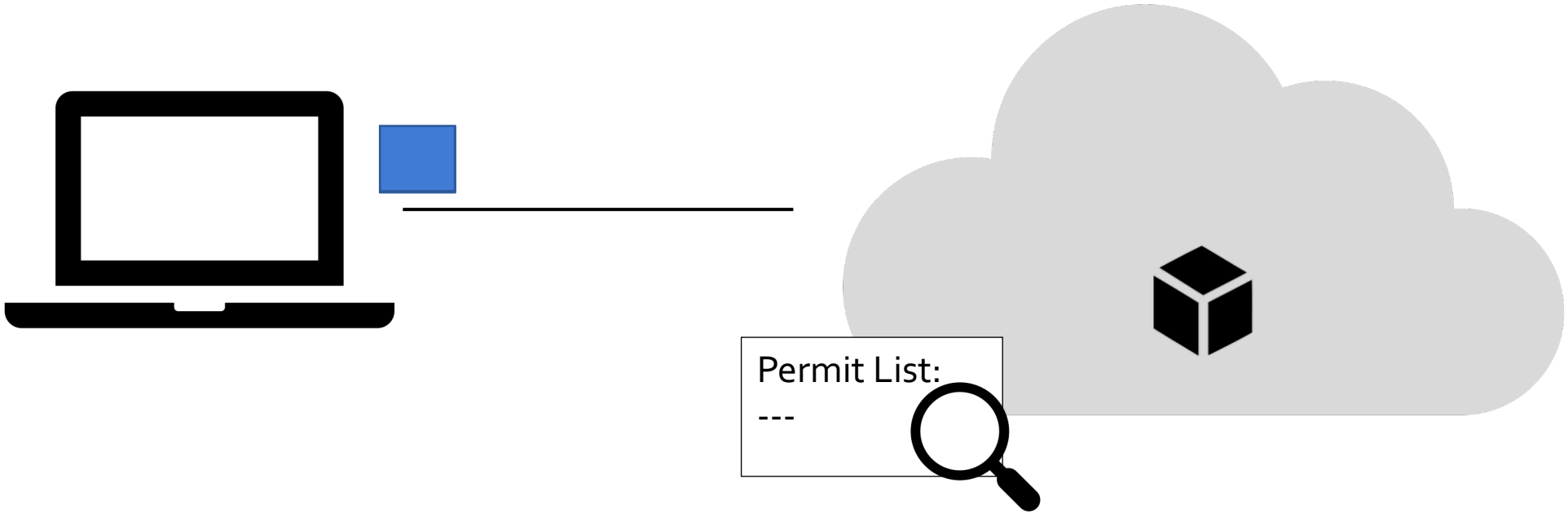
```
set_connectivity()
set_security()
…
```



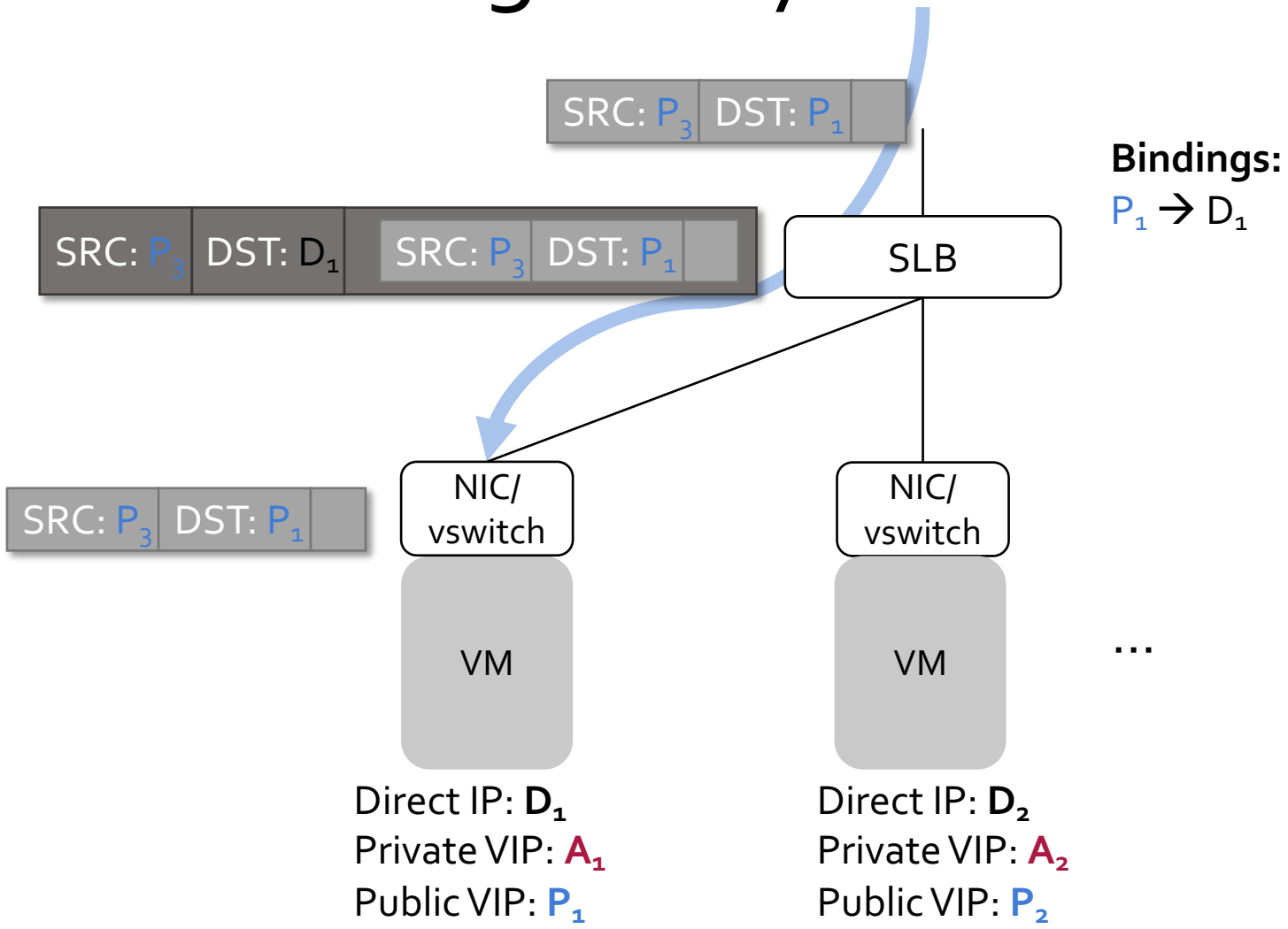
Connectivity Made Trivial

"Publicly routable but default-off"

PRDO

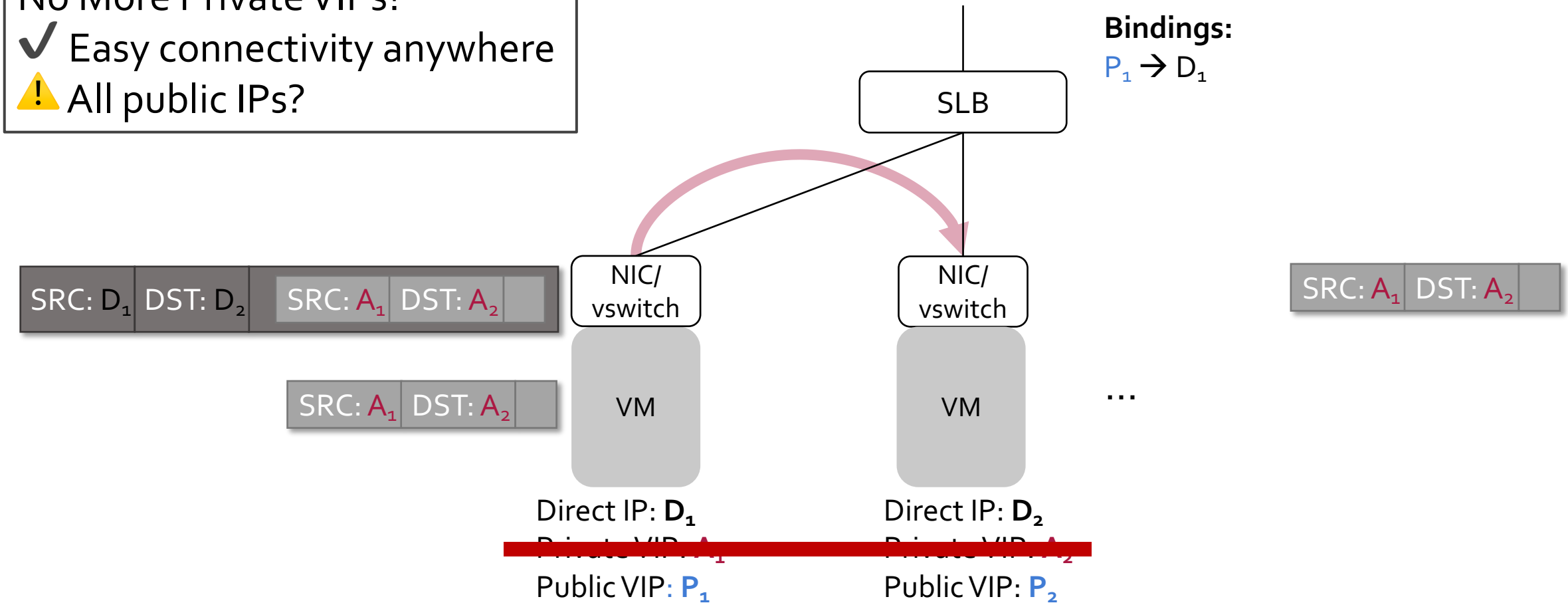


Cloud Addressing Today



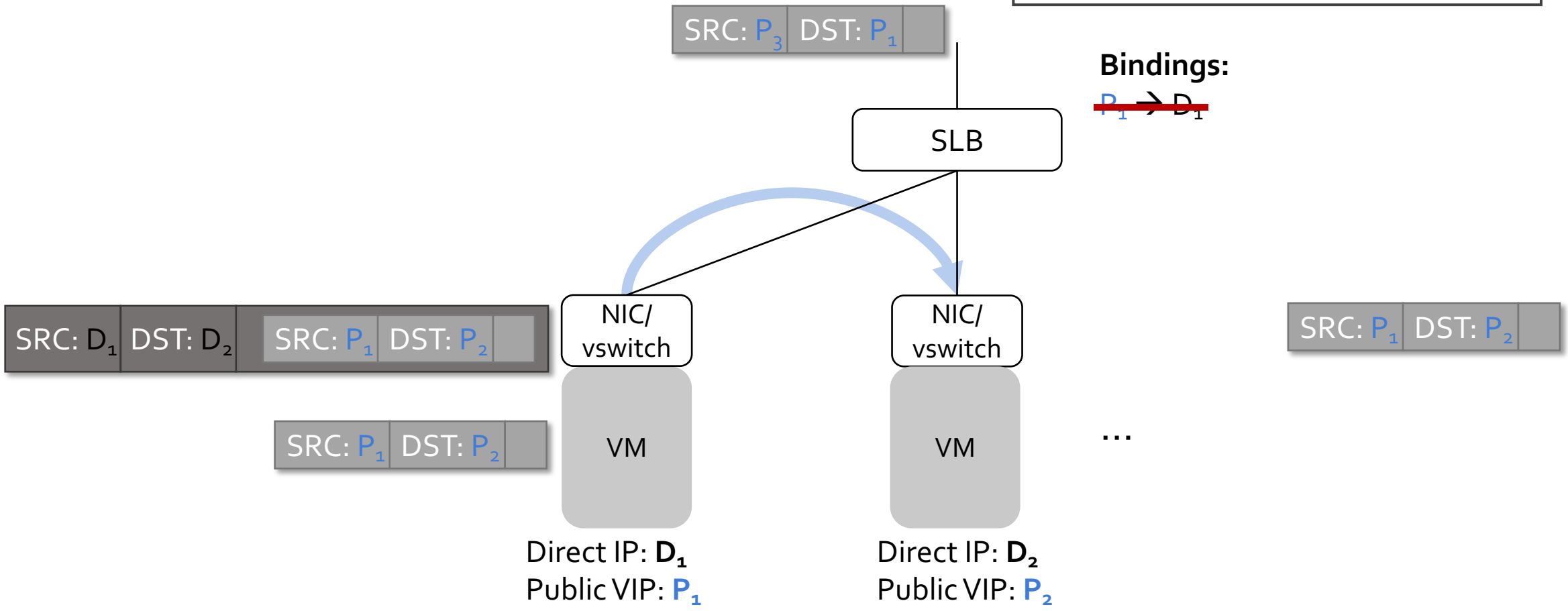
Cloud Addressing Today

No More Private VIPs?
✓ Easy connectivity anywhere
⚠ All public IPs?

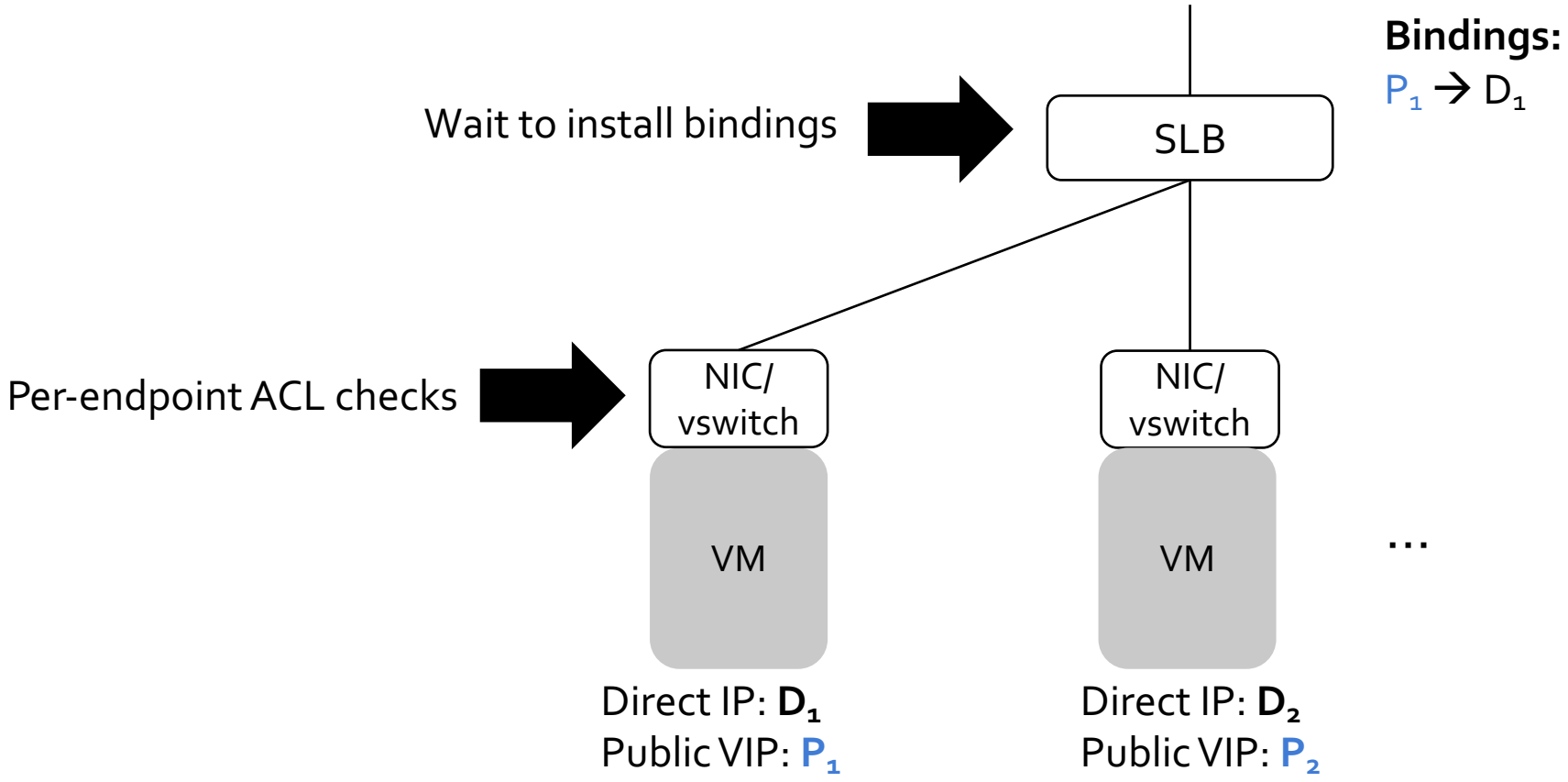


PRDO Addressing

Modification: Only install bindings when external connections permitted

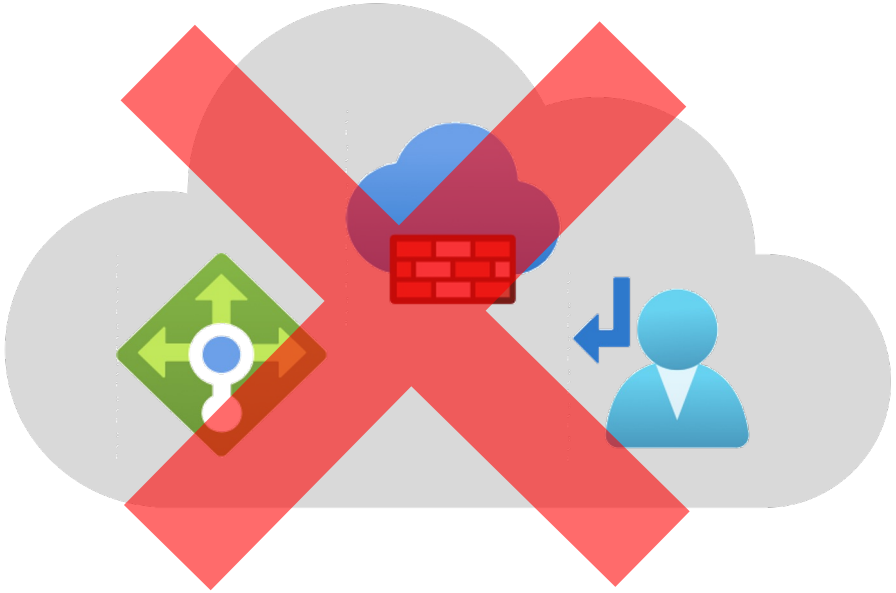


Connectivity Made Trivial

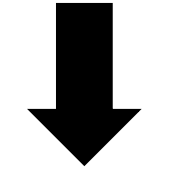


Approach

1. Make connectivity trivial
2. Design around tenant goals
3. Specify these goals on endpoints
4. Put the burden on the cloud provider

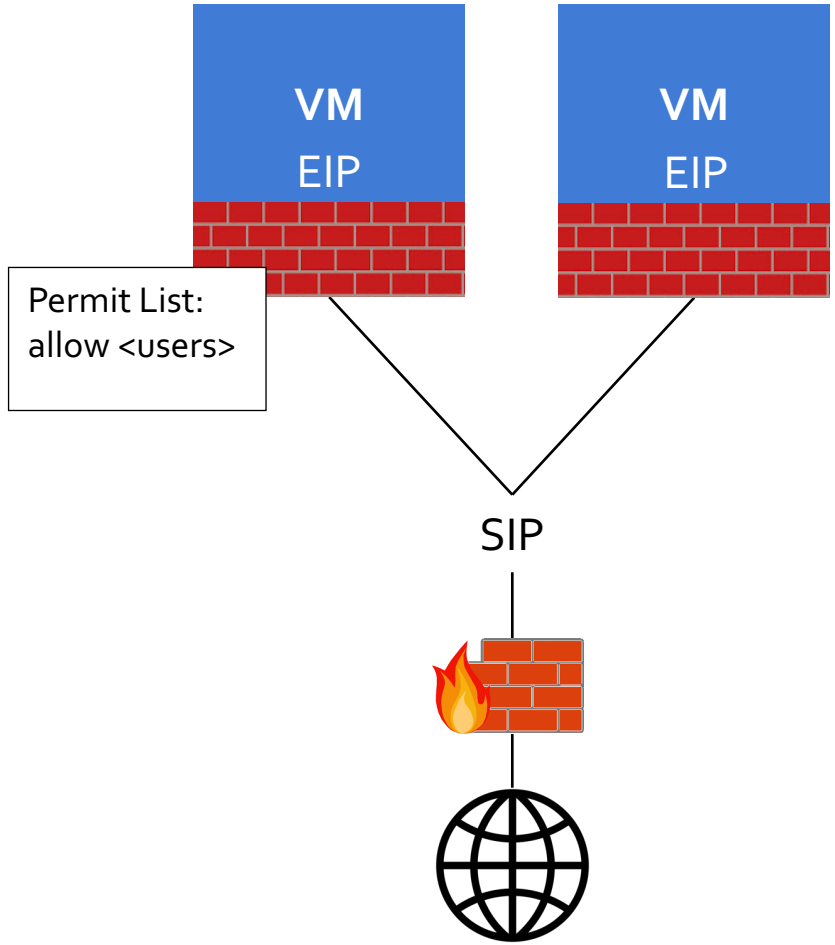


```
set_connectivity()
set_security()
…
```



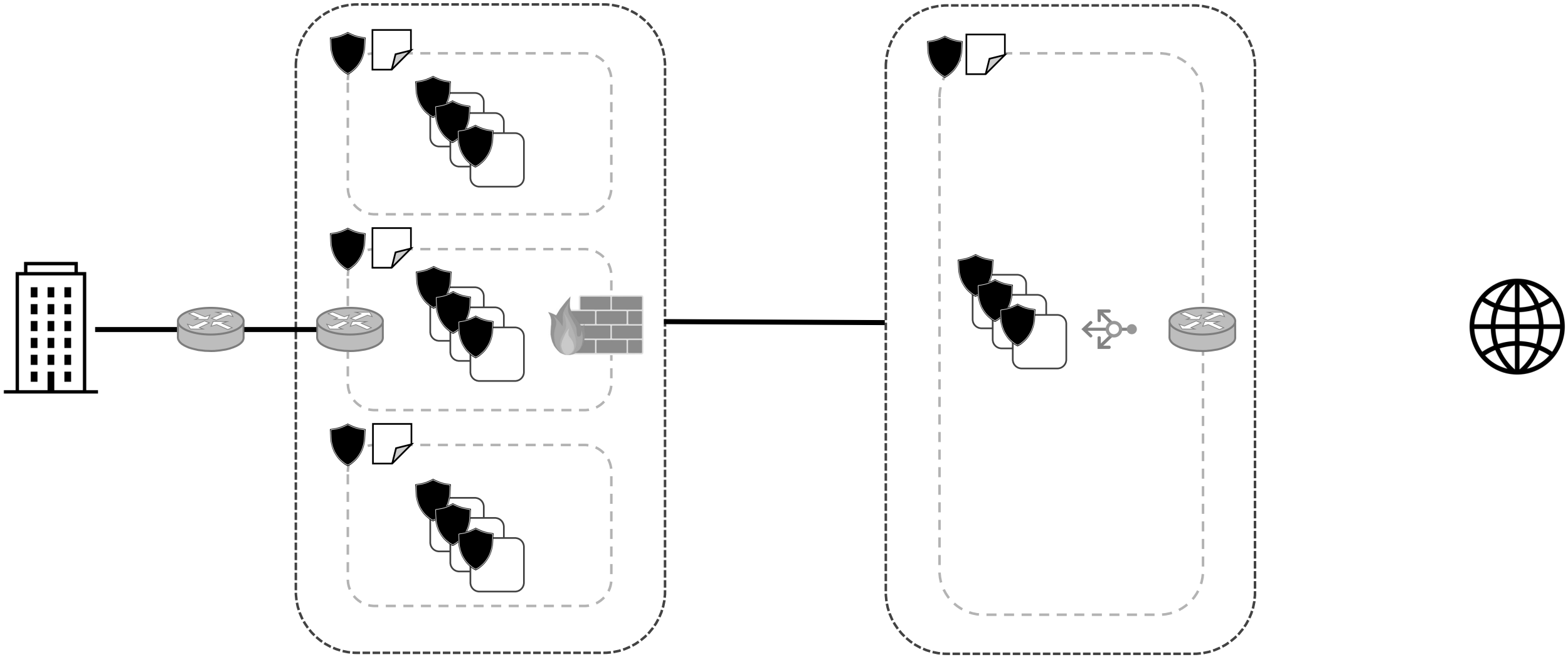
Tenant Goals → API

Connectivity:	eip = request_eip(vm_id)
Availability:	sip = request_sip() bind(sip, eip)
Security:	set_permit_list(eip, permit_list) annotate(eip, middlebox)
Performance:	set_qos(region, dst, bandwidth) set_qos_class(class, five_tuple)
Management:	set_tag(eip, tag)

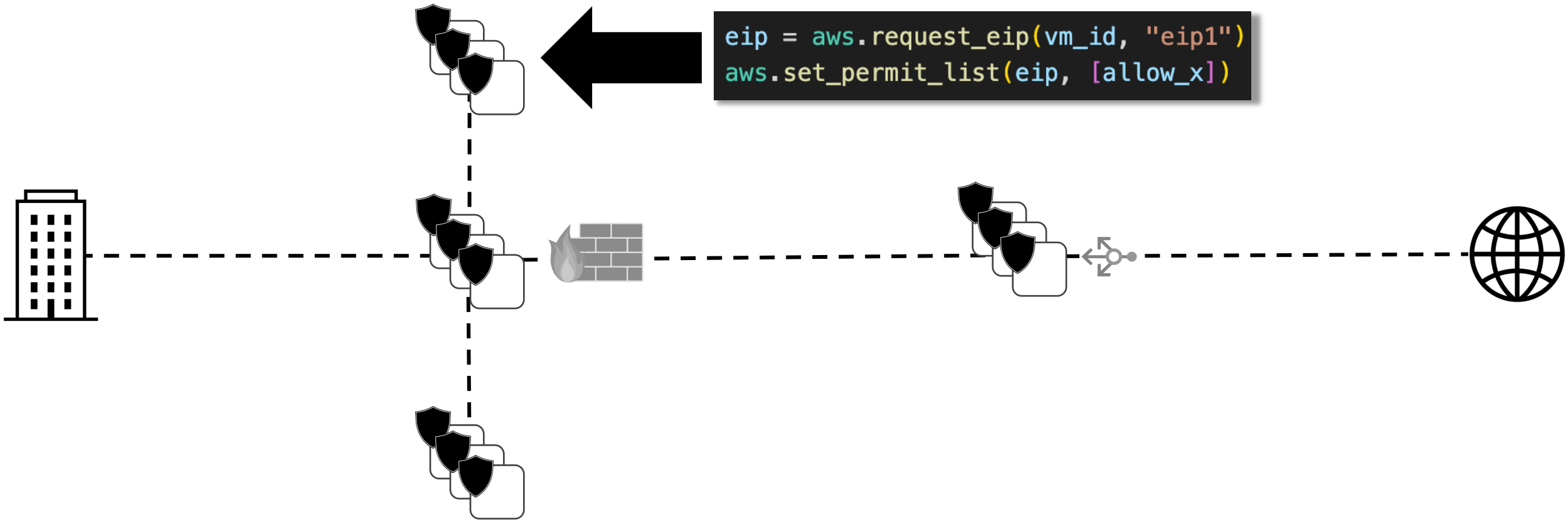


See the paper for more details

Invisinets in Action

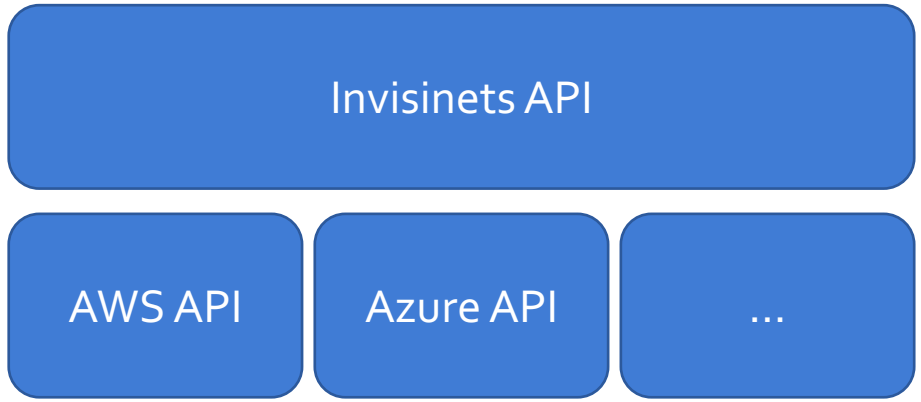


Invisinets in Action



Implementation

Now:



Long-Term:

- ✓ EIPs
- ✓ SIPs
- ✓ Permit lists
- ✓ Annotate
- ✓* QoS

Evaluation

- Scalability
  - Addressing Infrastructure
  - QoS Controller
- Simplicity
  - Case Studies
  - Terraform Scrape

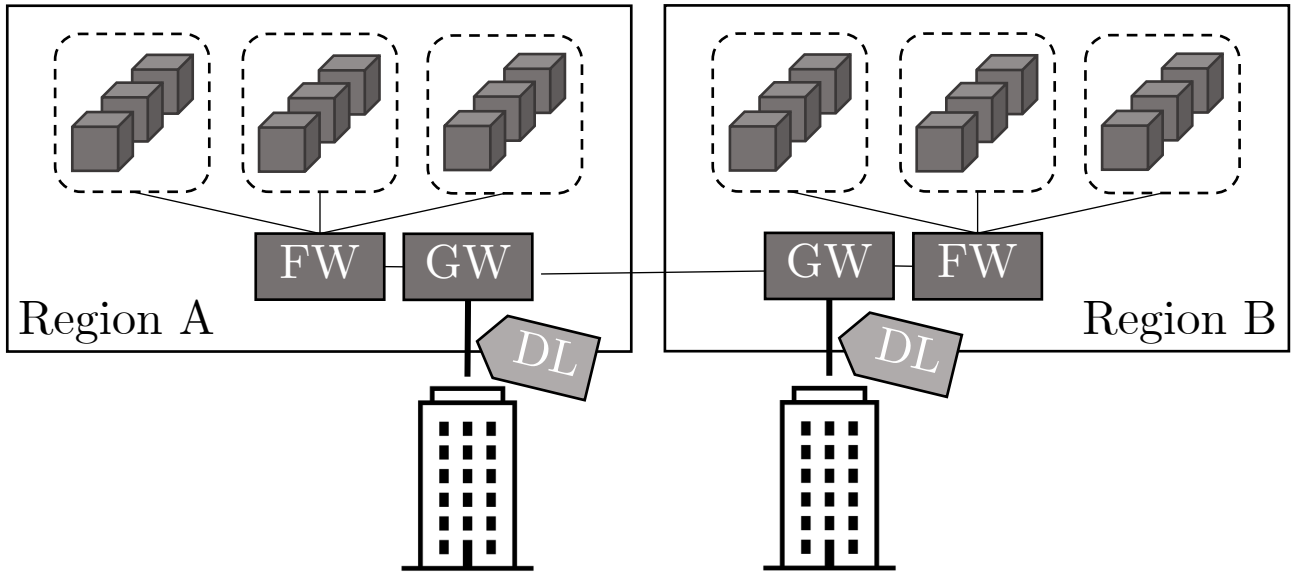
Evaluation

- Scalability
  - Addressing Infrastructure
  - QoS Controller
- Simplicity ⟵ **How do we measure this?**
  - Case Studies → **LoC, Boxes, Links, Points of Configuration***
  - Terraform Scrape → **Resources and LoC Removed**

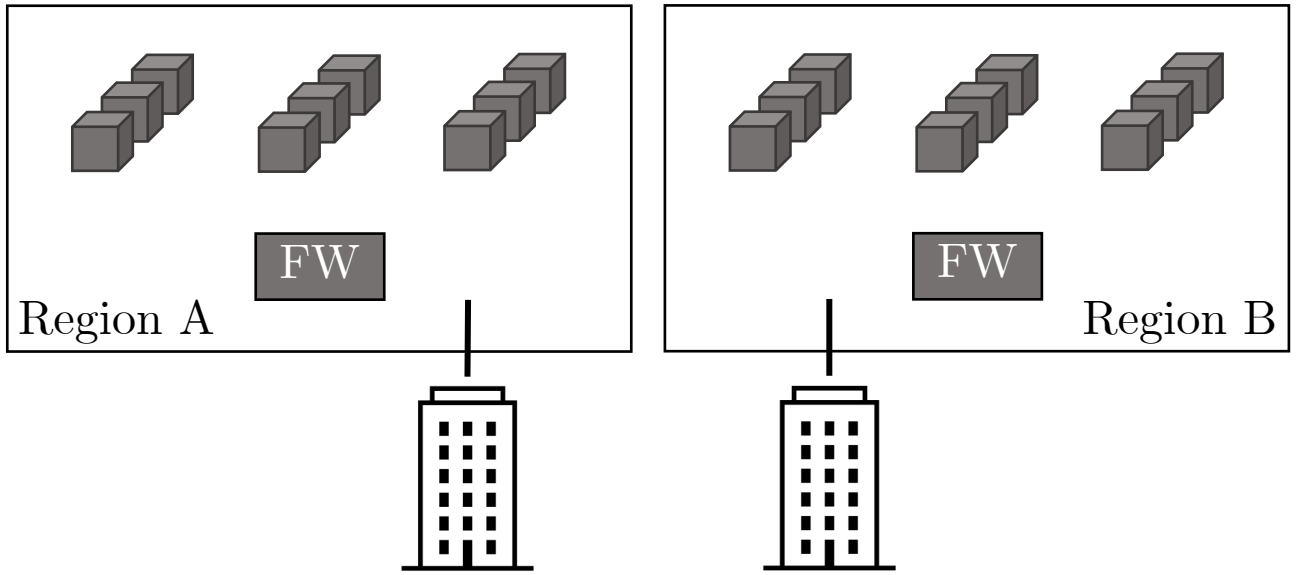
Check out the paper for additional case studies and evaluation

***ignoring endpoints and state that can be scaled arbitrarily**

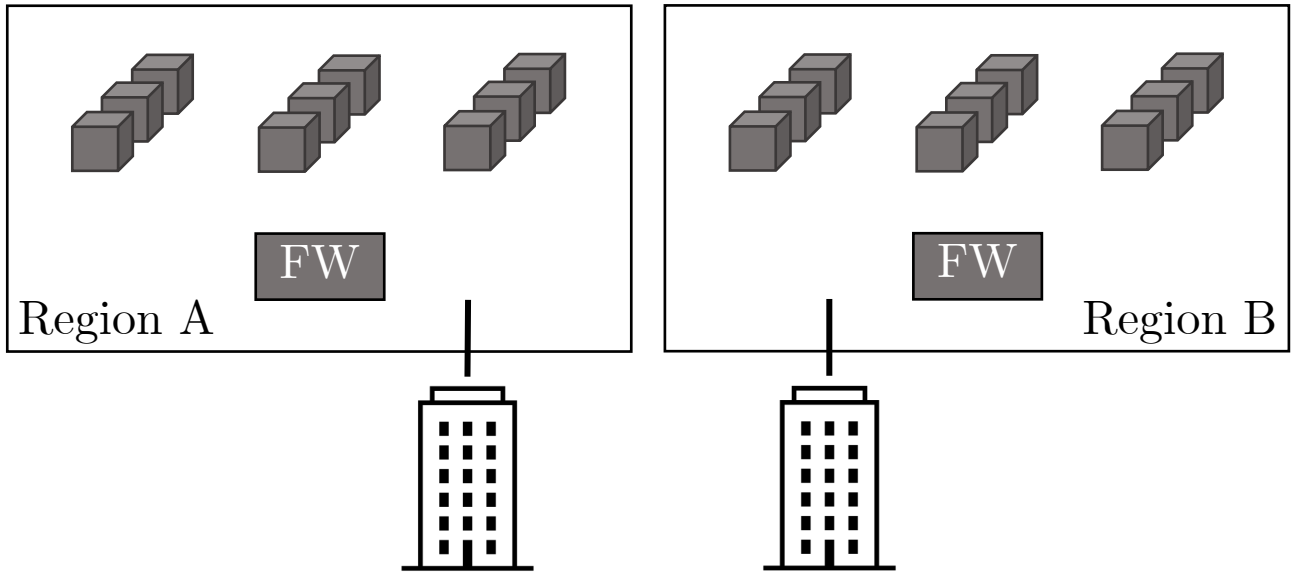
Case Study 1



Case Study 1

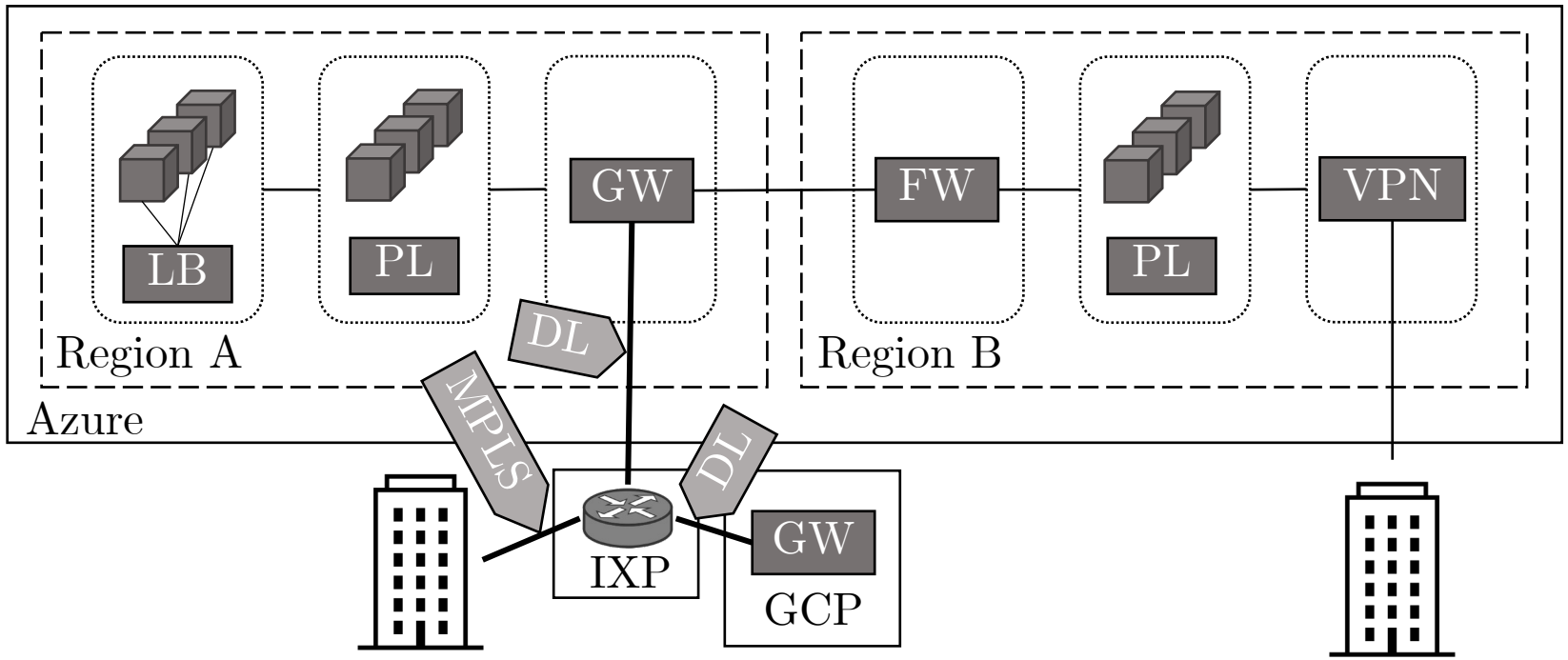


Case Study 1

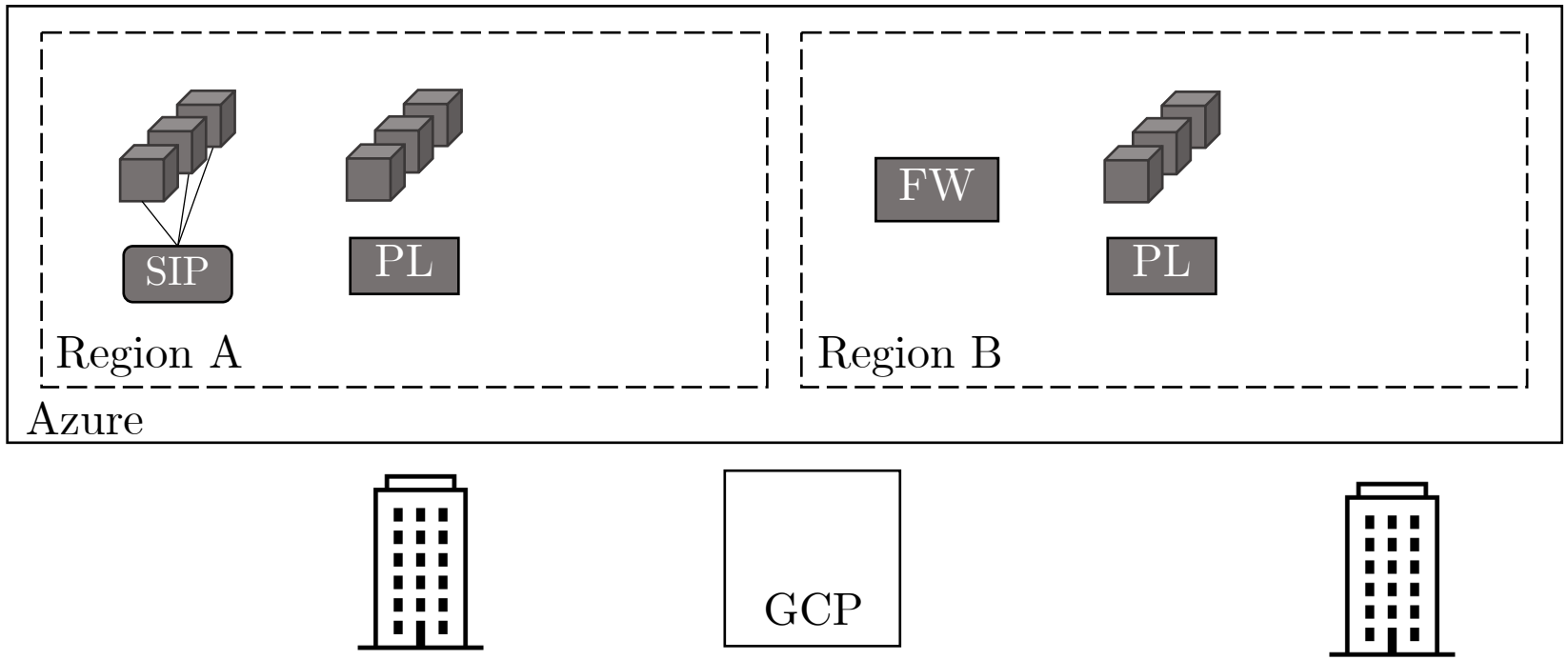


Metric	First-Party	Aviatrix	Invisinets
Boxes	6	18	1
Configuration Points	22	36	4
Links	5	2	0
LoC*	>45	~	4

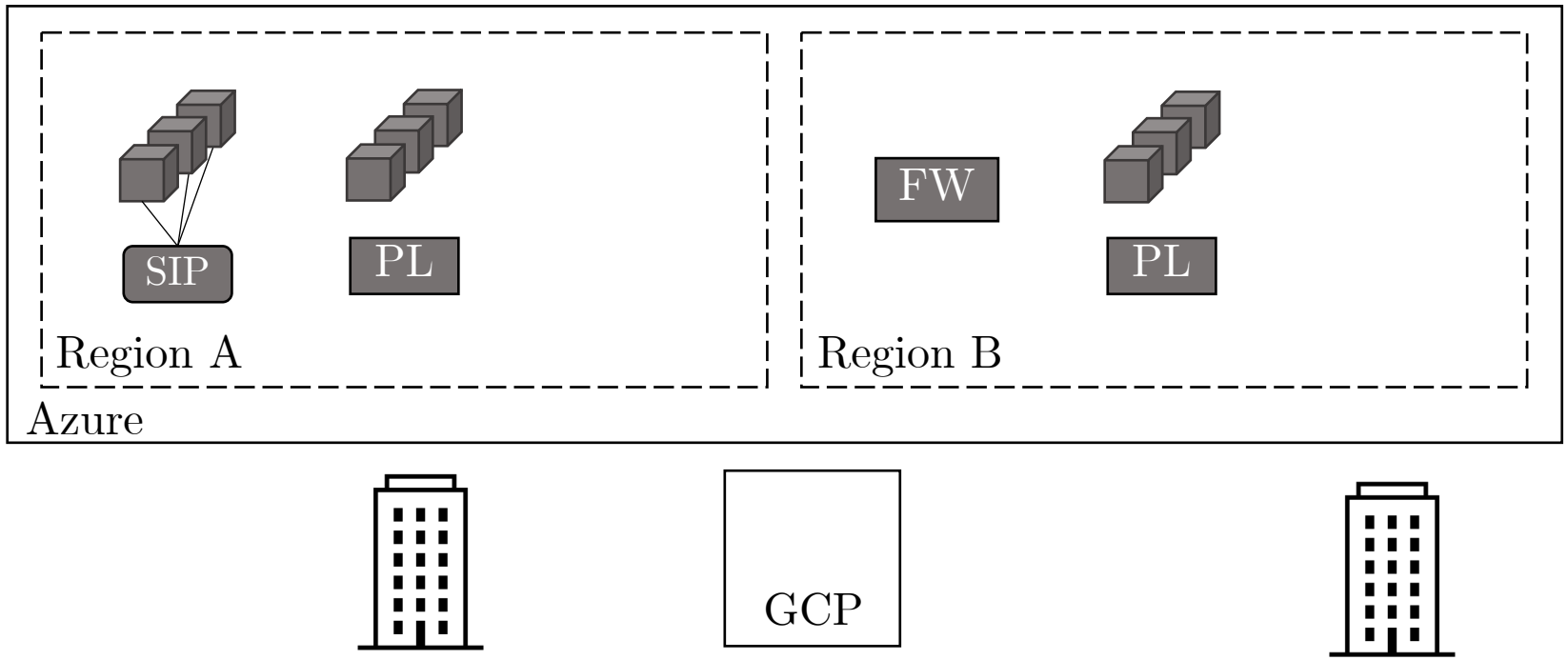
Case Study 2



Case Study 2



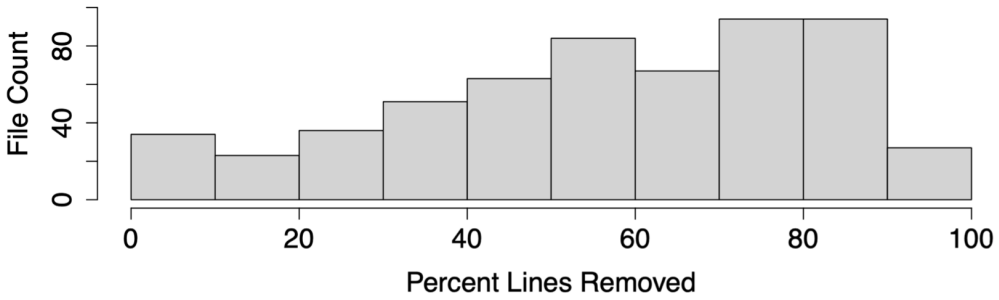
Case Study 2



Metric	First-Party	Aviatrix	Invisinets
Boxes	6	6	1
Configuration Points	22	22	4
Links	5	5	0
LoC*	>80	~	4

Terraform GitHub Scrape

Virtual Network Component	Occurrence Count	Line Count
VPC	2,493	26,731
Route Table	1,839	13,317
Subnet	1,514	14,677
Security Group	456	9,704
Internet Gateway	445	2,802
Route	339	2,184
NAT Gateway	209	1,661
Network ACL	141	2,841
Transit Gateway	82	587
VPN Gateway	40	316
Load Balancer	22	207
Network Interface	16	123
VPN Peering Connections	5	27
Customer Gateway	4	58
VPN Route	2	10
VPN Connections	1	13



Conclusion

- **Declarative** API defined on **endpoints**
- Makes connectivity trivial with **PRDO** addressing
- Up to **90% decrease in complexity**

See our code at https://github.com/smcclure20/invisinets

sarah@cs.berkeley.edu