
AWS
re:Invent

FSI302

Security at scale: How Goldman Sachs manages network and access control

Robert Cossin
Vice President
Goldman Sachs

Sujoy Saha
Associate
Goldman Sachs

Harsha W Sharma
Solutions Architect
AWS

AWS re:Invent

© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Today's discussion

Who are we?

Core security tenets

Initial approach

TxB micro-account approach

Connecting to endpoint services cross-account

Connecting to endpoint services from on-premises

Lessons learned



© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Who we are

Goldman Sachs (GS)

- Leading global investment bank

Transaction banking (TxB)

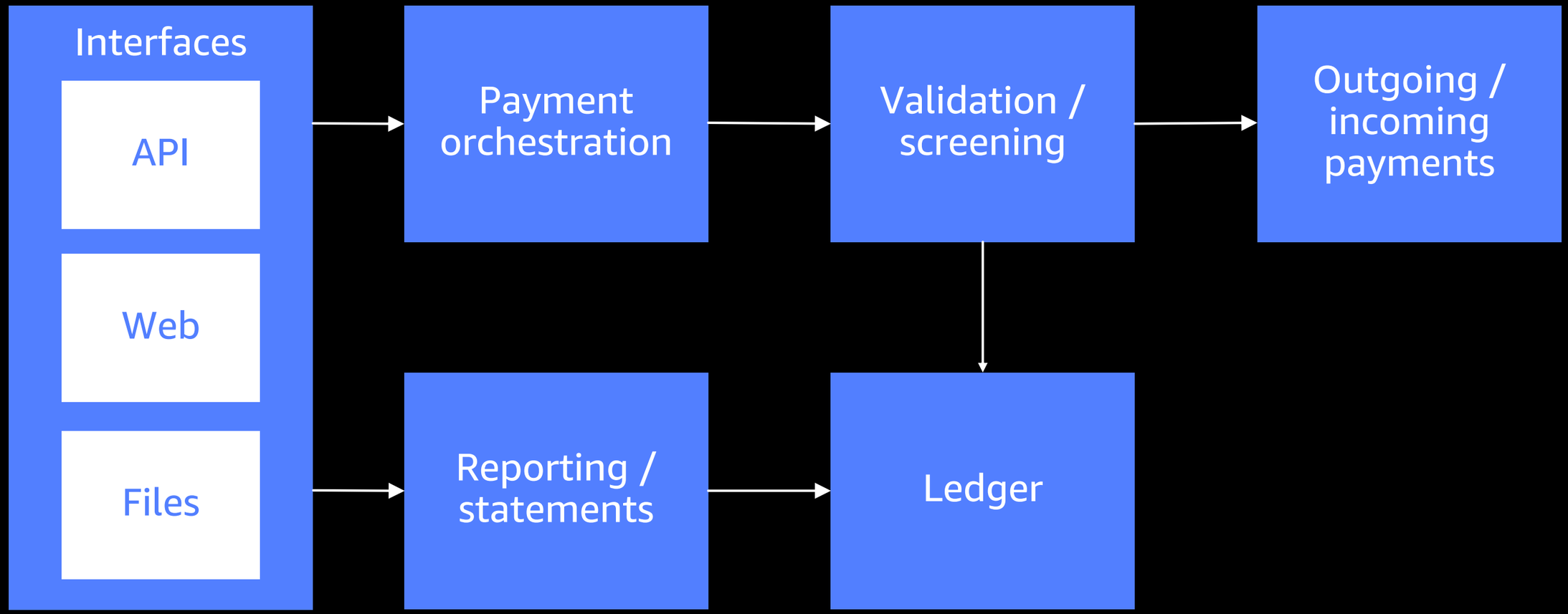
- Provides cash management services to corporate clients, including deposits, payments, and analytics
- Most components built on AWS

GS engineering

- Public cloud engineering team
- Focus on cloud governance and security



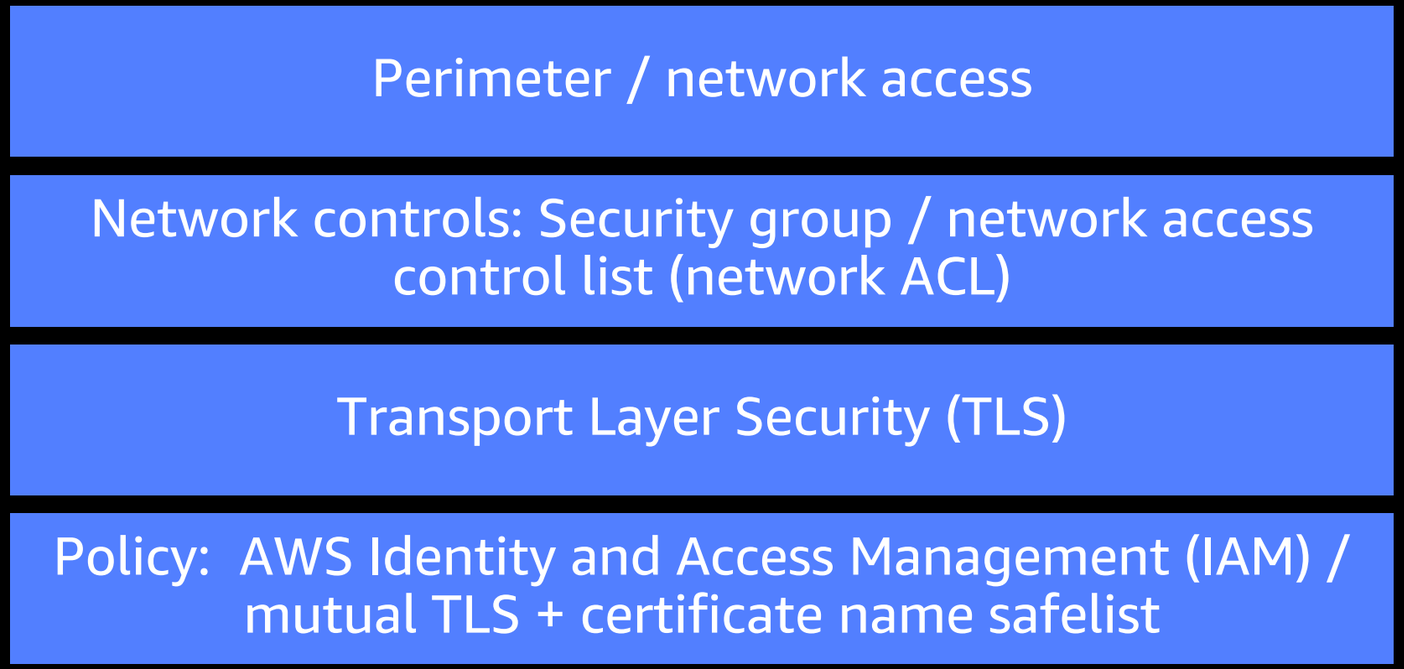
High-level functional bank components



TxB's core security tenets

Implement security at all levels to ensure defense in depth

Ensure breach of or mistake in one layer does not compromise stack



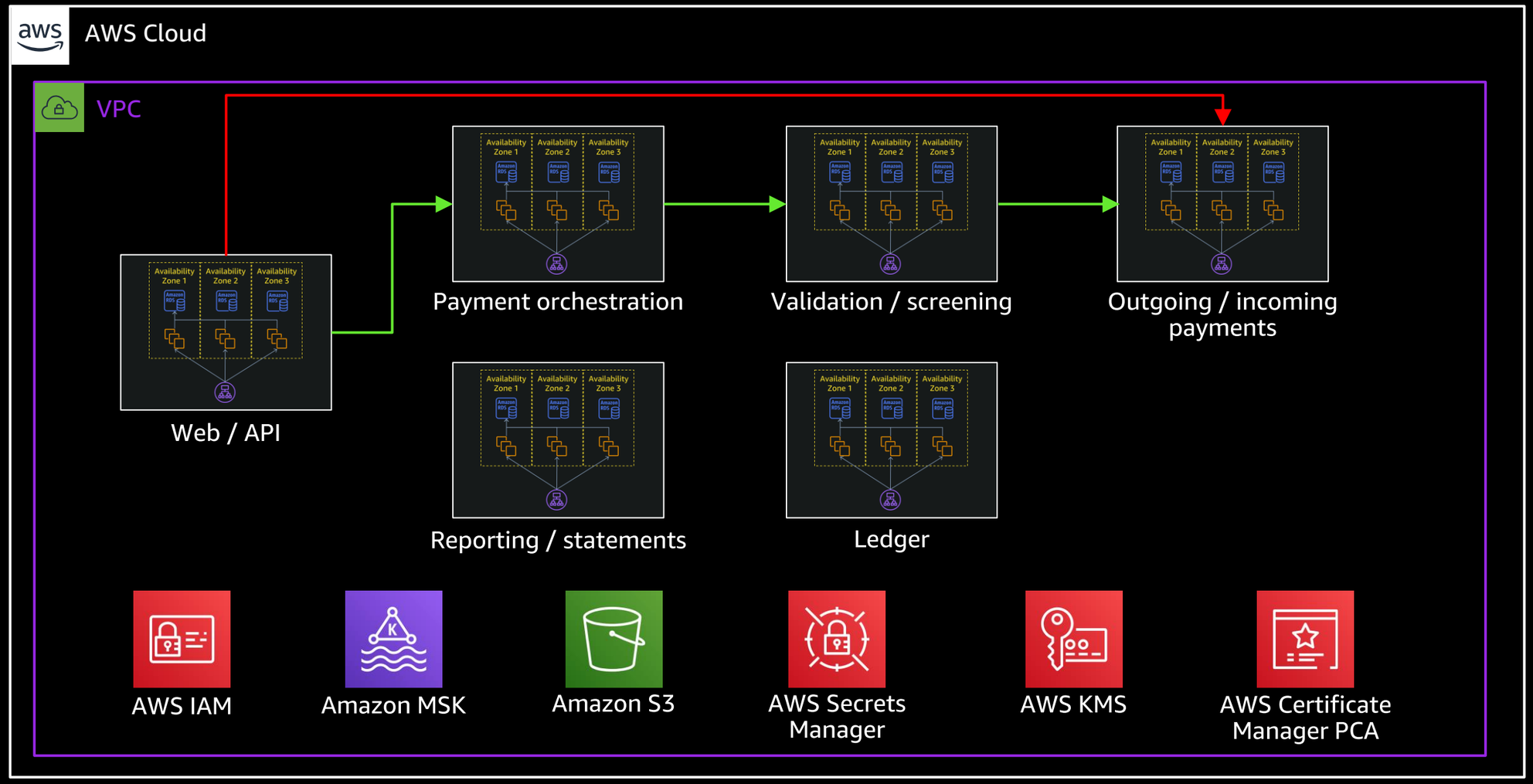
Initial approach



© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Initial approach: Single large TxB account

How can we ensure service connectivity is tightly controlled?
How can we minimize data and / or resource leakage?



Initial multi-tenant approach: Challenges

Perimeter and network controls

- Security group and network ACL management

Scaling limits

- Amazon Virtual Private Cloud (Amazon VPC) IP space
- AWS quotas: Load balancers, Amazon Elastic Compute Cloud (Amazon EC2) instances, IAM roles, etc.

Policy

- Fine-grained IAM and resource-based policies
- Complexity can lead to unintended configuration errors

Identity vs. resource policies

You can define IAM and resource policies – they are complementary

Identity-based policy	Resource-based policy		
	Allow	No allow / deny	Deny
Allow	Explicitly allowed	Explicitly allowed	Explicitly denied
No allow / deny	Explicitly allowed	Implicitly denied	Explicitly denied
Deny	Explicitly denied	Explicitly denied	Explicitly denied

* A notable exception is AWS Key Management Service (AWS KMS), where both identity and resource allow are required

AWS documentation on how identity-based policies and resource-based policies are evaluated together:

https://docs.aws.amazon.com/IAM/latest/UserGuide/access_policies_identity-vs-resource.html

Resource policy for a Lambda function

```
{
  "Version": "2012-10-17",
  "Id": "default",
  "Statement": [
    {
      "Sid": “allow-lambda-role",
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::123456789101:role/invoke_lambda_role"
      },
      "Action": "lambda:InvokeFunction",
      "Resource": "arn:aws:lambda:us-east-2:123456789101:function:sample_lambda"
    }
  ]
}
```

So only “invoke_lambda_role” can invoke the Lambda, right?

IAM policy is created for a different role

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "",
            "Effect": "Allow",
            "Action": "lambda:InvokeFunction",
            "Resource": [
                "arn:aws:lambda:us-east-2:123456789101:function:sample_lambda",
                "arn:aws:lambda:us-east-2:123456789101:function:sample_lambda_2"
            ]
        }
    ]
}
```

So how do we prevent this? Ensure all policies include appropriate “deny” statements. Can we ensure that every team always creates valid policies?

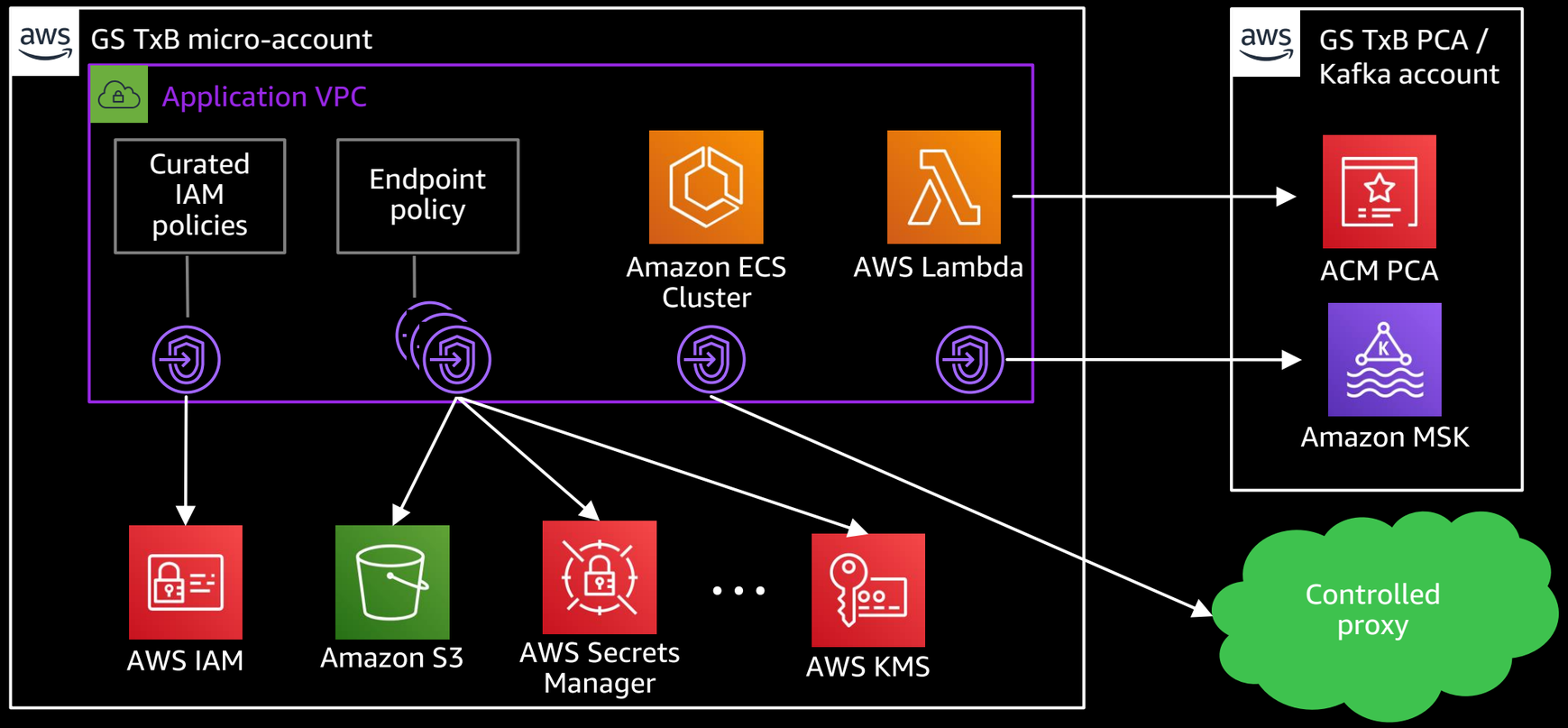
TxB micro-account approach



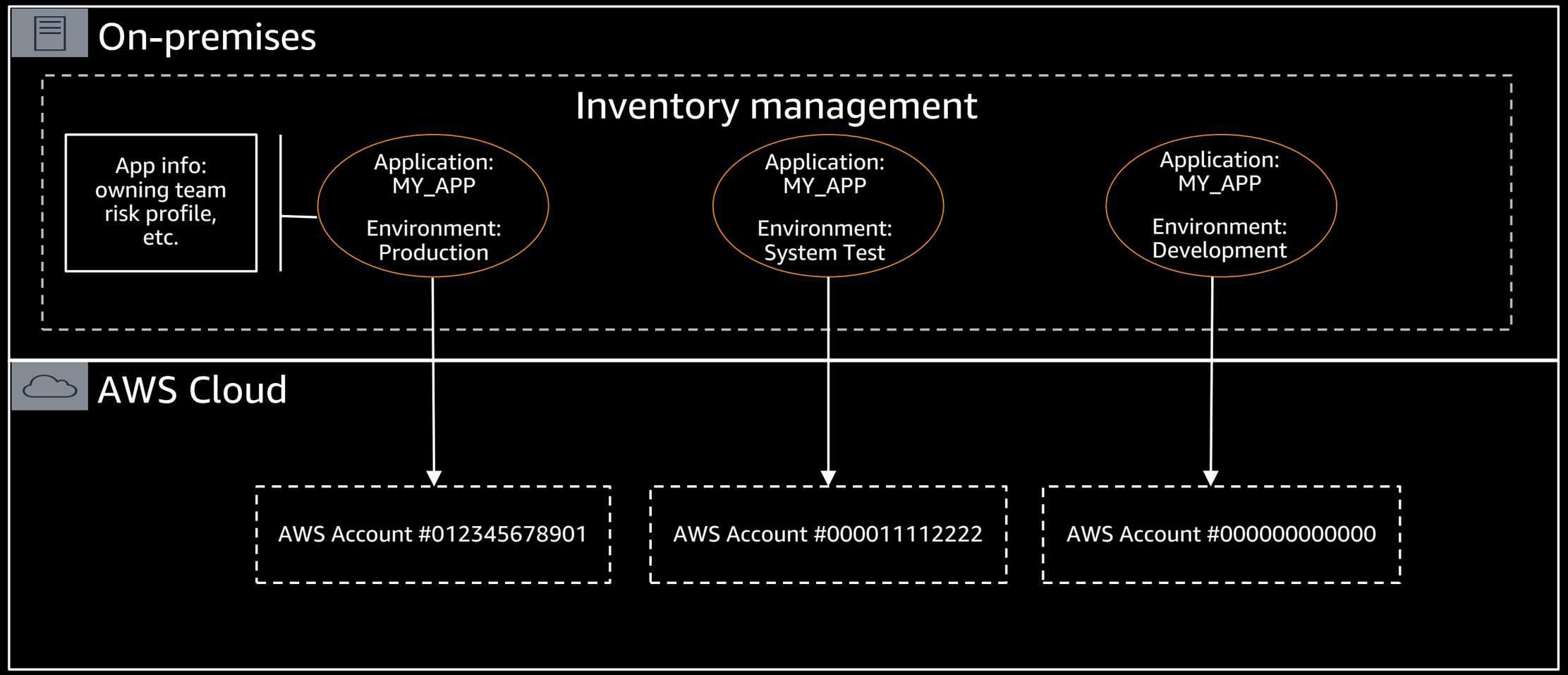
© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Micro-account: GS AWS account-per-service

Everything you need to build an application and connect to other applications



Micro-account: GS AWS account-per-service



Connecting to endpoint services cross-account



AWS re:Invent

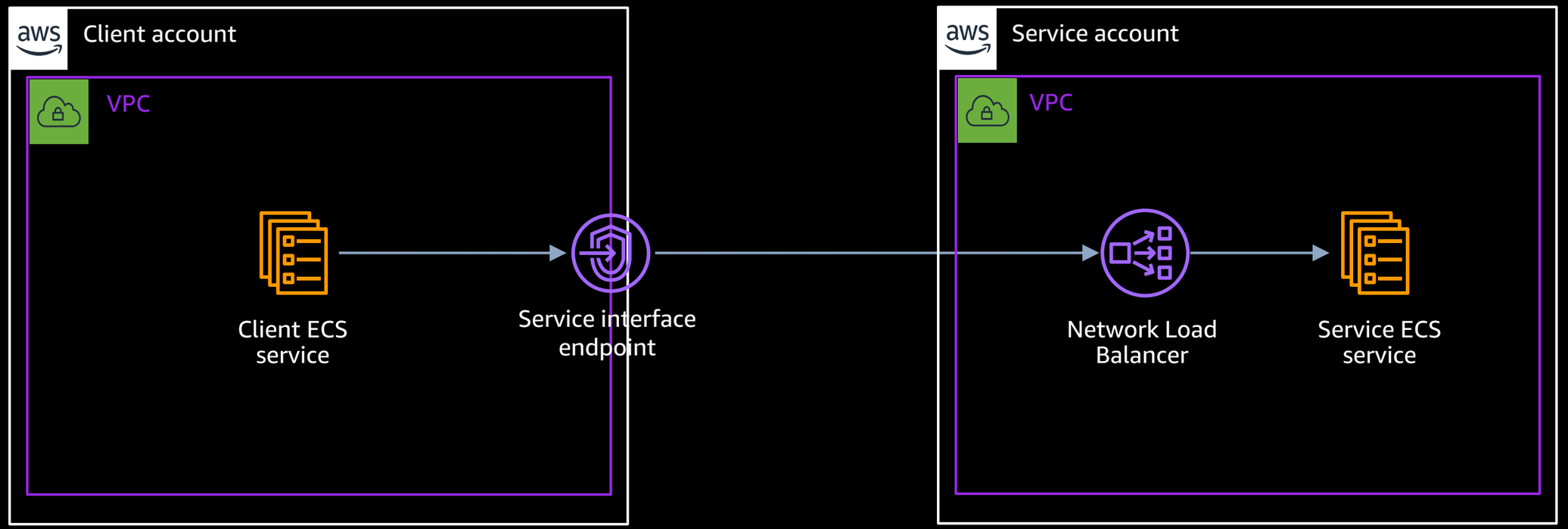
© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Micro-account: Network connectivity options

Expansion strategy	Network flow direction	Perimeter control	Limits
Peering	Bidirectional	Security groups and network ACLs	Accounts and VPC limits; must manage IP Classless Inter-Domain Routing (CIDR) carefully
VPC sharing	Bidirectional (same VPC)	Security groups and network ACLs	VPC limits
AWS Transit Gateway	Bidirectional	Security groups and network ACLs + route tables	Accounts and VPC limits; must manage IP CIDR carefully
AWS PrivateLink	Unidirectional	Fine-grained: Only specific service is shared	Accounts and VPC limits; CIDR overlap permitted

We chose PrivateLink because we liked the perimeter control / network segmentation and required bilateral agreement for each connection

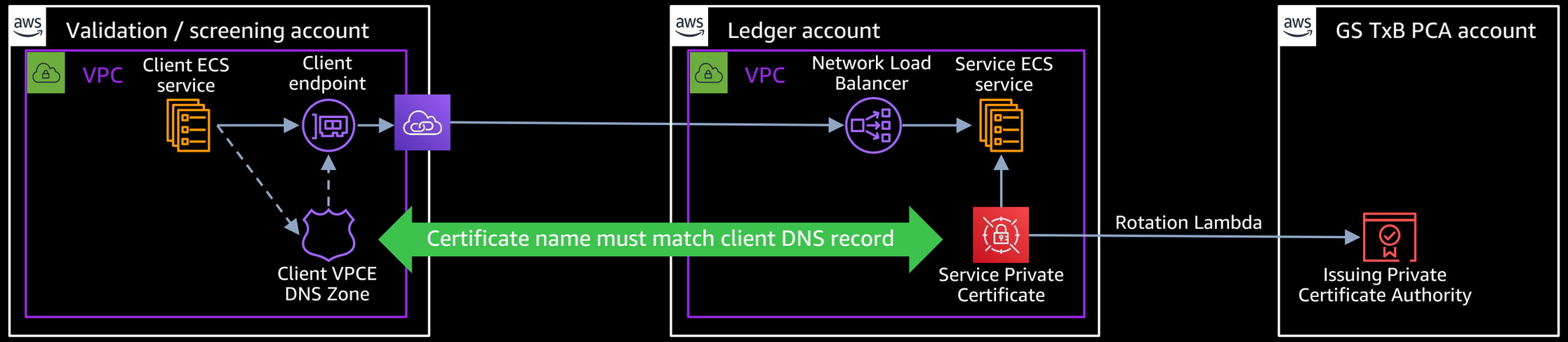
AWS PrivateLink in TxB



VPC endpoint services in TxB

Client and service must be in sync with respect to:

- Domain Name System (DNS) record: Required for TLS handshake
- Availability Zones (AZs): Endpoints must be on same AZs as service
- Ports: Clients must set up appropriate security groups

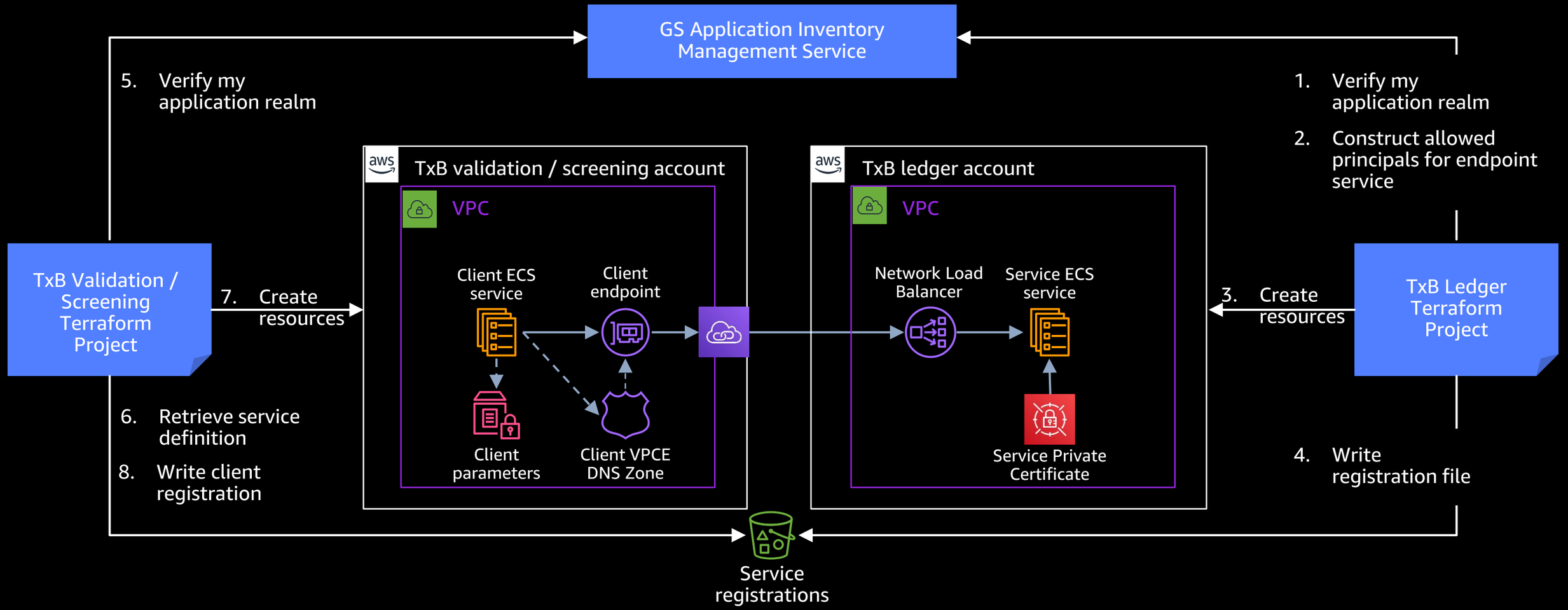


- How do we keep services and clients in sync?

VPC endpoint management in TxB

Amazon Simple Storage Service (Amazon S3)–based service registry maps logical service name to AWS service name

Ensures connections only between TxB accounts within same application realm



Connecting to endpoint services from on-premises



AWS re:Invent

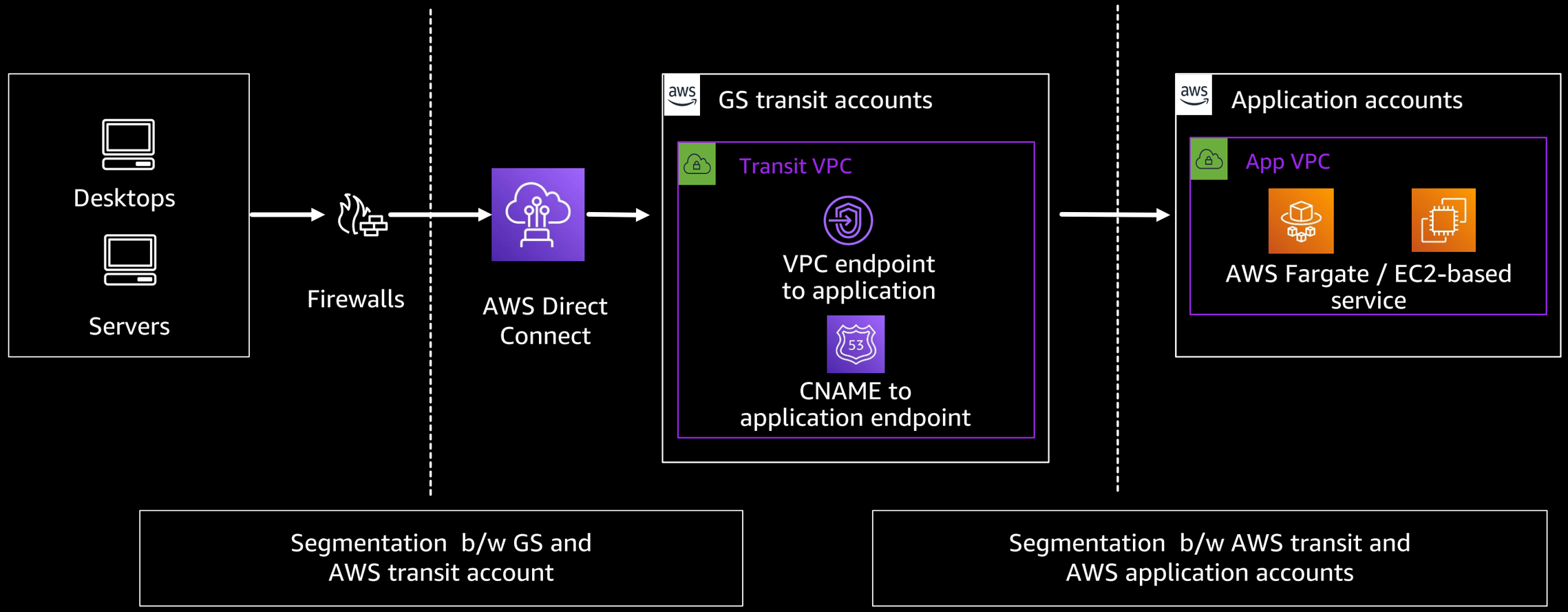
© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.

GS on-premises to AWS connectivity

Many options were considered

- AWS Direct Connect / Transit Gateways to Application VPCs
  - Difficult to manage subnets over thousands of AWS accounts under different ownership
  - GS on-premises network extended to application VPCs
- Go over the internet
  - Application VPCs need to be exposed to internet
- Use a transit VPC
  - Control egress / ingress to different applications using PrivateLink
  - No subnet management overhead for teams
  - Strong separation of application VPCs from GS network

GS transit VPC: Connectivity and segmentation

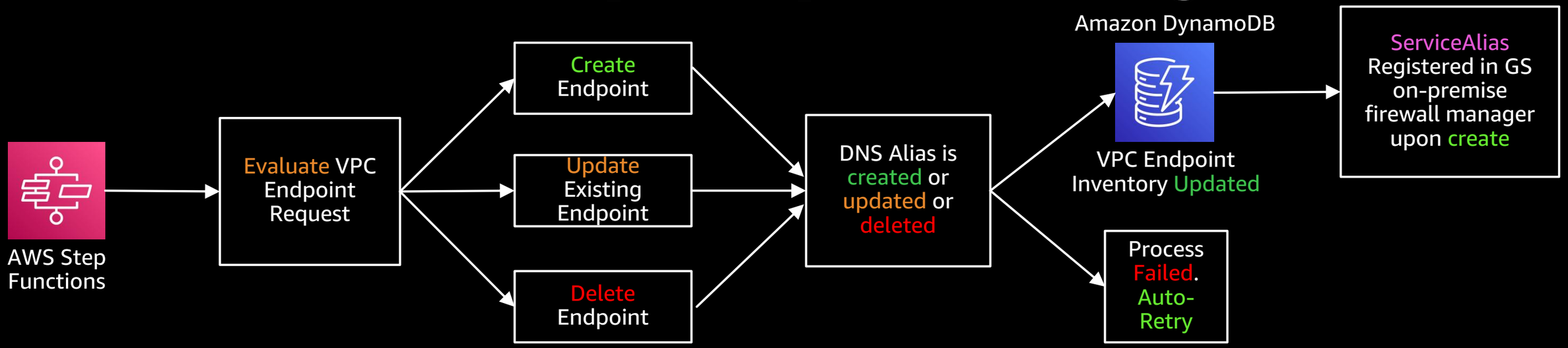


Automatic VPC endpoint provisioning



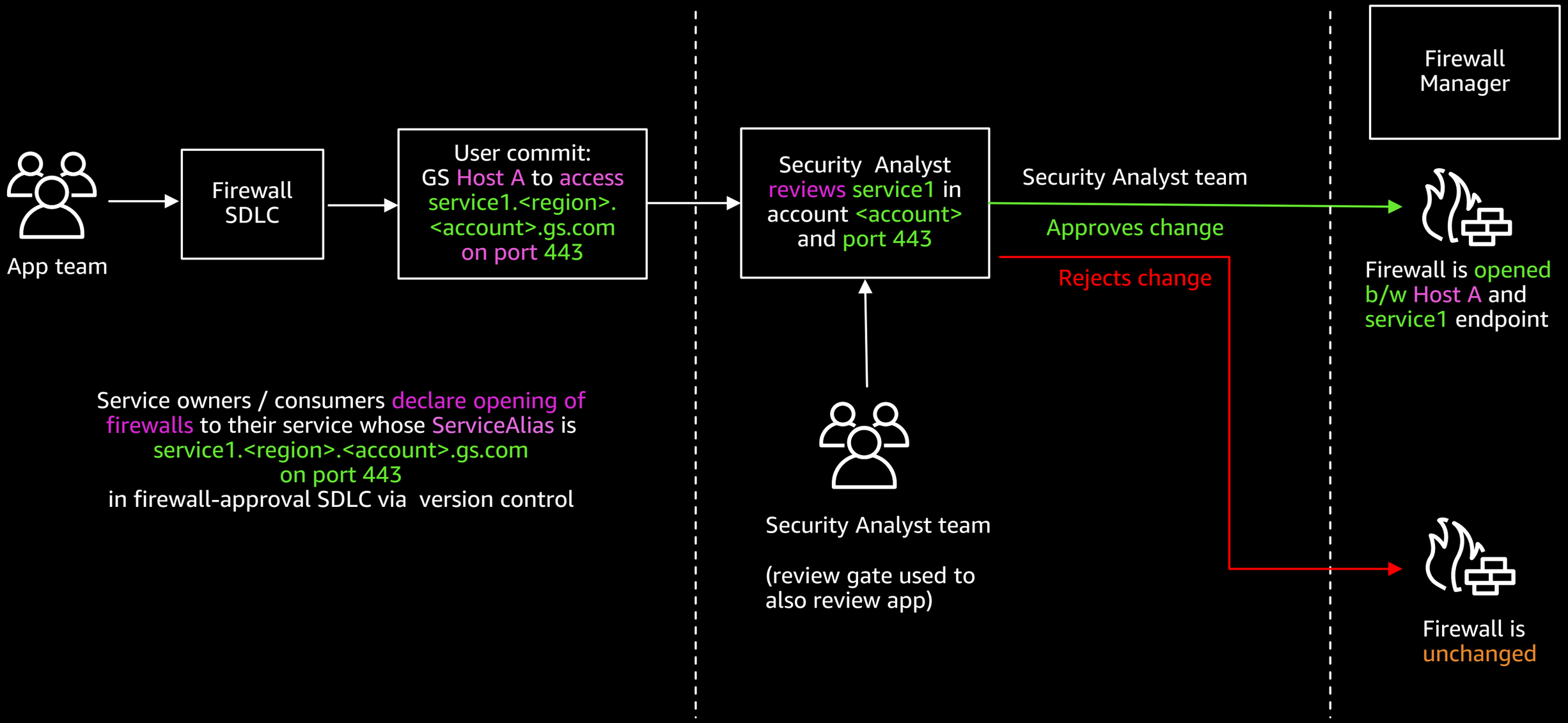
- App team declares intent to connect to AWS-hosted application via IAC modules (Terraform or AWS CloudFormation)
- App team will create:
  - VPC endpoint service, Network Load Balancers, and appropriate application targets
  - A custom Amazon Route 53 SRV record with details such as
    - Friendly Application Name (my-gs-app-1)
    - Ports (443)
    - Availability Zone IDs (use1-az1, use1-az2)
    - VPC endpoint service name (com.amazonaws.vpce.us-east-1.vpce-svc-some-service-name)
- These SRV record creates/updates/deletes are forwarded to VPC endpoint provisioning step functions

Automatic VPC endpoint provisioning



- The automatic endpoint provisioning service will verify
  - VPC endpoint request is compliant with our security standards
  - GS app account is in good standing with the GS AWS organization
- Endpoint will be provisioned after all validations are passed
  - A human readable DNS Alias is created in Route 53 (default endpoint DNS assigned by AWS is not)
  - This alias, ServiceAlias, remains constant for any create/update/delete endpoint request
  - Example: service1.<region>.<account>.gs.com, which is a CNAME of vpce-xx1.vpc-svc-xx2.amazonaws.com
  - The endpoint is tracked in inventory
- The ServiceAlias is registered with the GS firewall manager on-premises for new services

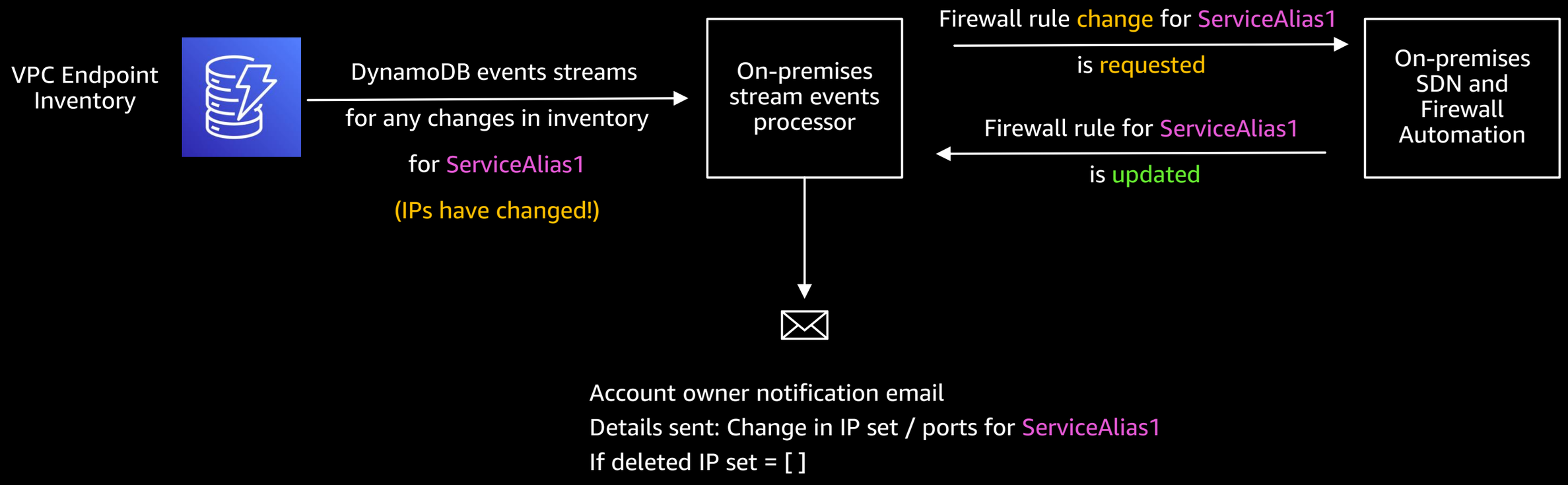
Automatic firewall updates: Requesting access



Automatic firewall updates: Changes in IP

An app owner has a ServiceAlias1: service1.<region>.<account>.gs.com and it had IPs [10.1.1.11, 10.1.1.12]

If VPC endpoint service is deleted and re-created: IP Set changes to [10.1.2.11, 10.1.2.12] as IPs are auto-assigned by AWS



Lessons learned

- Account boundaries and use of AWS PrivateLink strongly improves security posture but there are some challenges
- Drive good user behavior and adoption of best security practices by:
  - Providing reusable infrastructure-as-code modules.
  - Align best practices with smoothest development path.
  - Utilize automation to minimize developer effort.
- Allow business units to experiment and innovate within guardrails, maintaining communication with core engineering.

Thank you!



AWS re:Invent

© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.



Please complete the session survey

AWS re:Invent

© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.