AWS Multi-Account Strategy

Old World IT

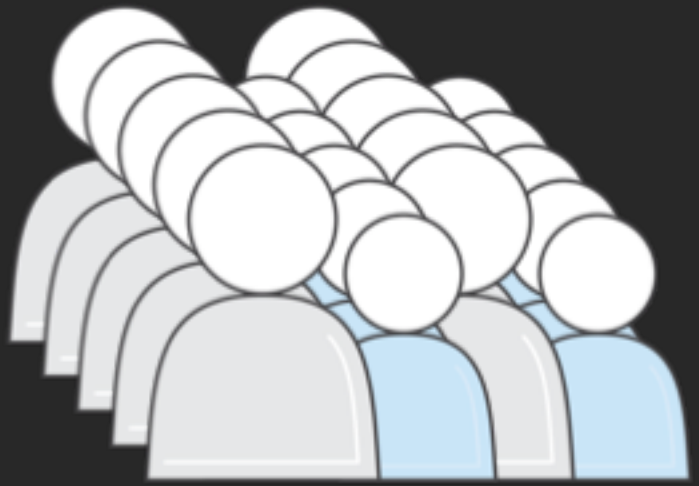
Bob – IT/security guy

Developers

Old World IT - Scale

More Bobs

More developers

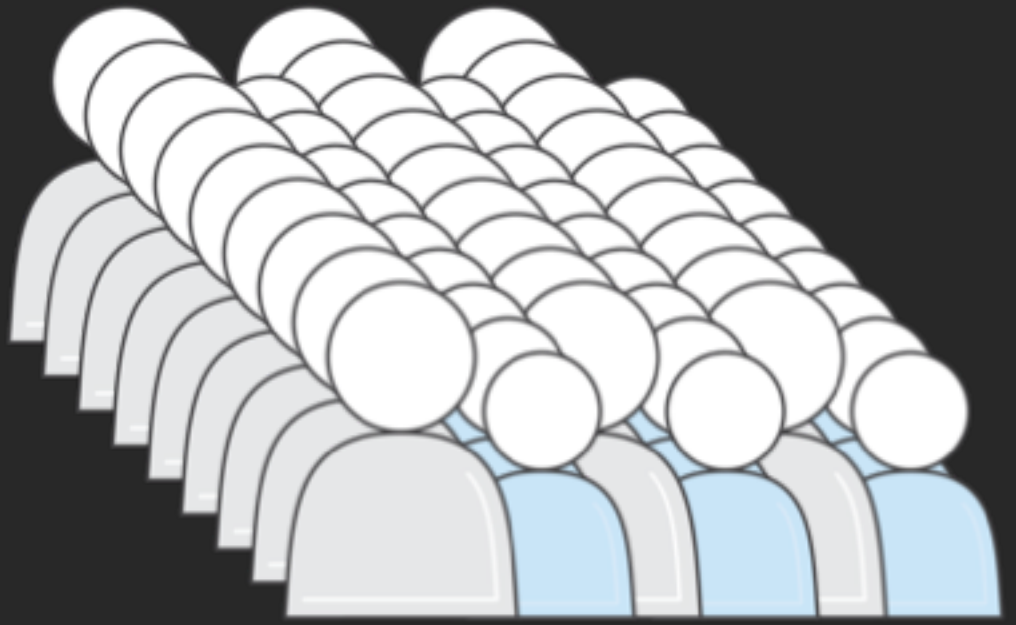


aws

The cloud will make this easier!

Same Bobs

More developers!



aws

One account, Isolation with IAM and VPC

Everything

"Gray" boundaries

Complicated and messy over time

Difficult to track resources

People stepping on each other

Separate developer account

Still can't track resources or spend
Still have isolation and blast radius concerns
Developers still stepping on each other
Bob now has to manage IAM and VPCs, here too

Dev

Prod

The problem

On-premises posture for the cloud

Inheriting ideas from datacenter days

Management and Ops don’t trust dev with full access

Developers want to work – Really!

DevOps is a great idea
    Doesn’t work when Ops is in the way

aws

A New Solution – We need

Access to AWS services without barriers

Ability to fail fast without collateral damage

Smaller blast-radius

Operations team → Cloud architects

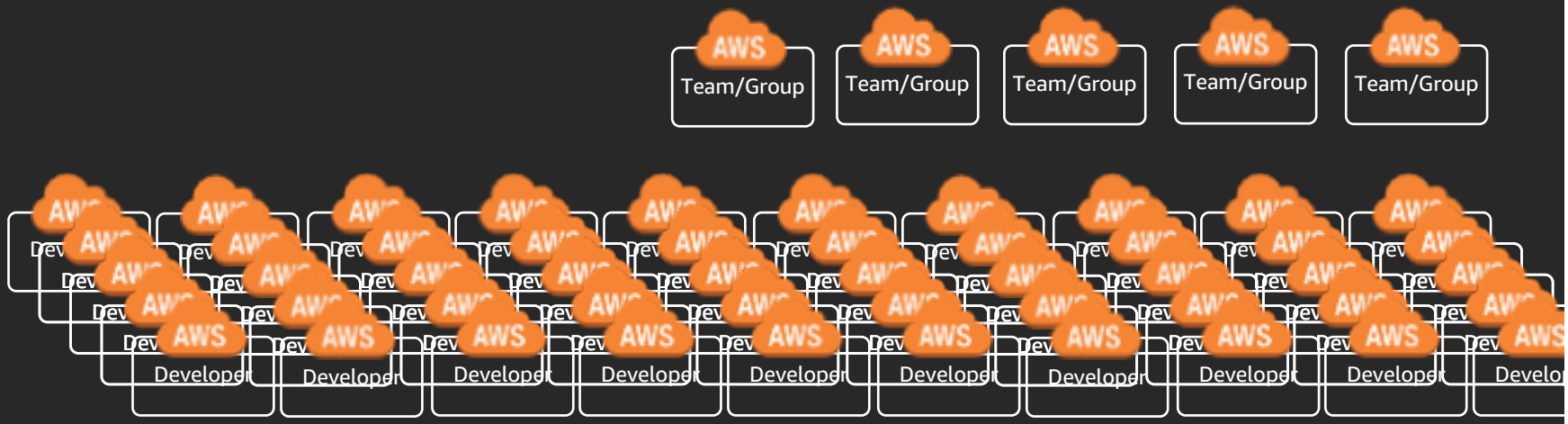
Everyone able to influence digital transformation

Costs and resources tracked to individuals and teams

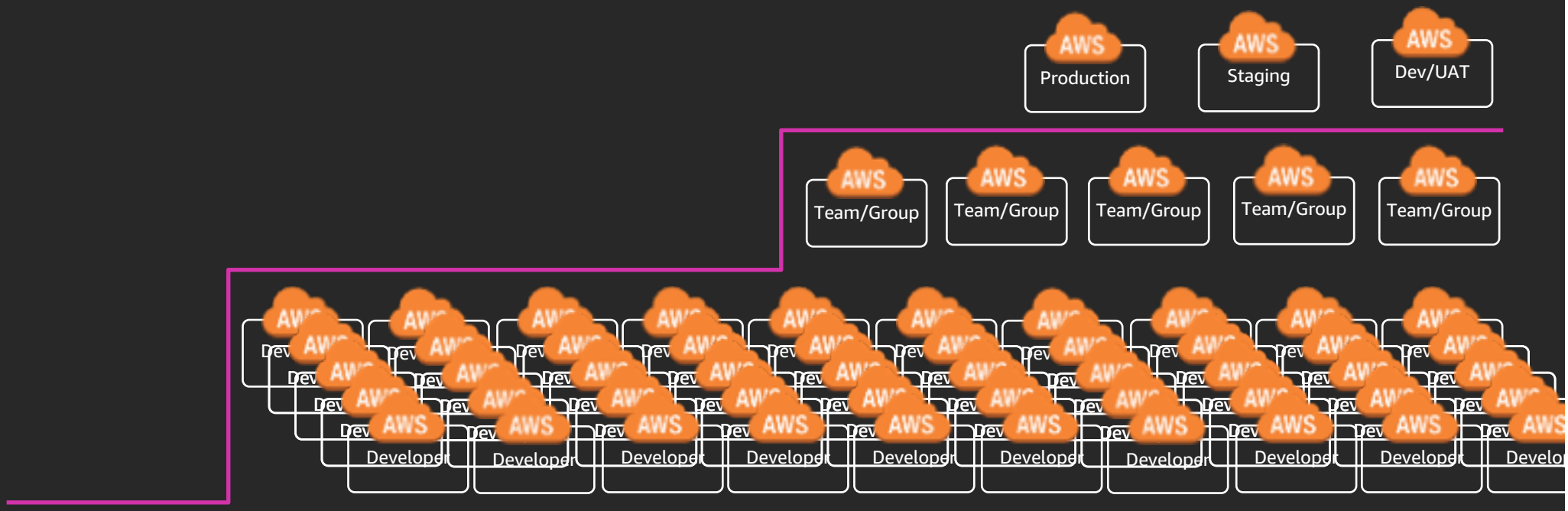
Optimize code for AWS

Where Do I Start? Developer accounts

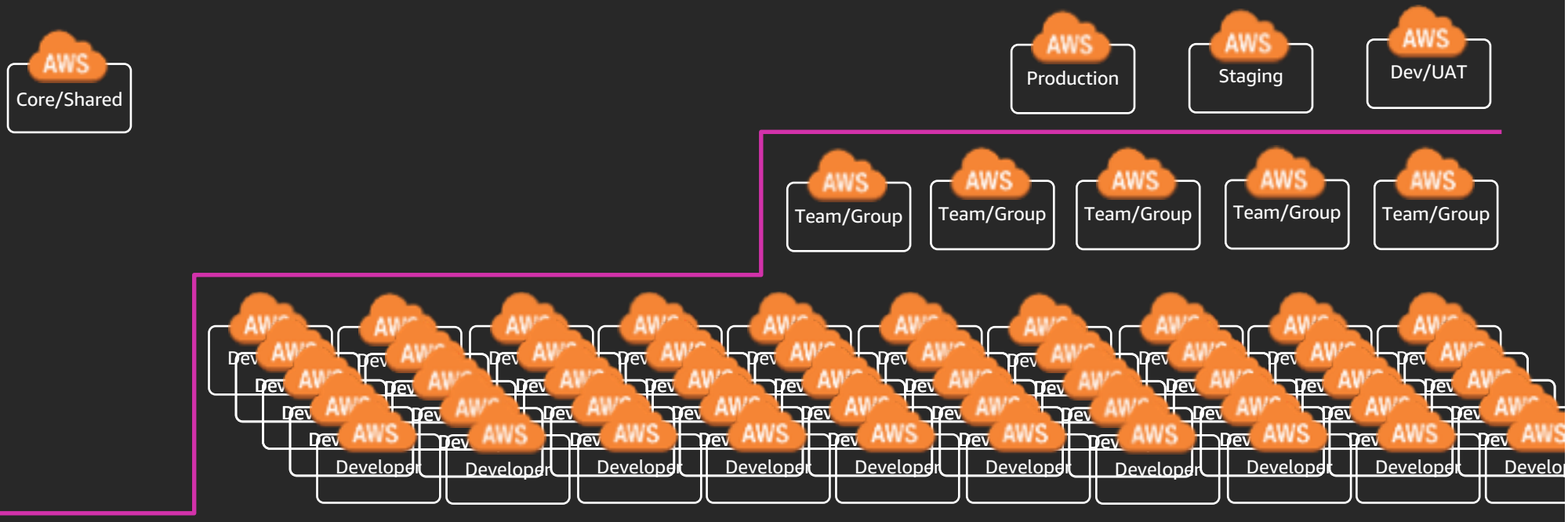
Where Do I Start? Team accounts



Where Do I Start? Ops accounts



Where Do I Start? Shared services



What are core shared accounts?

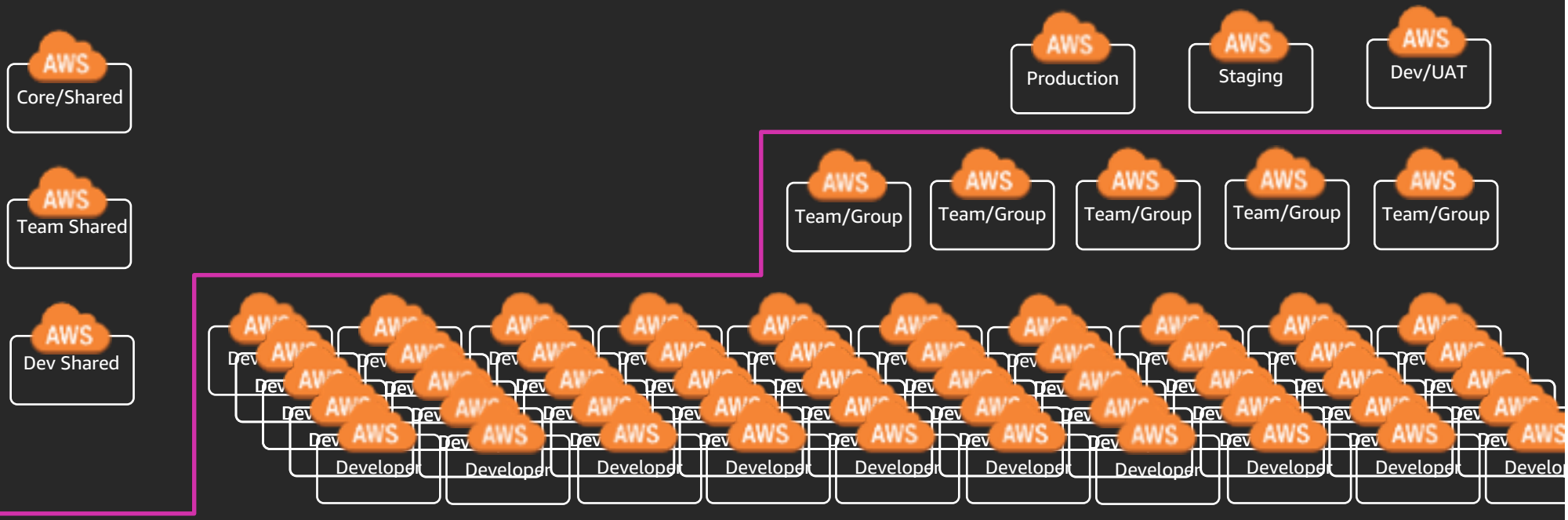
Shared Services

Log Archive

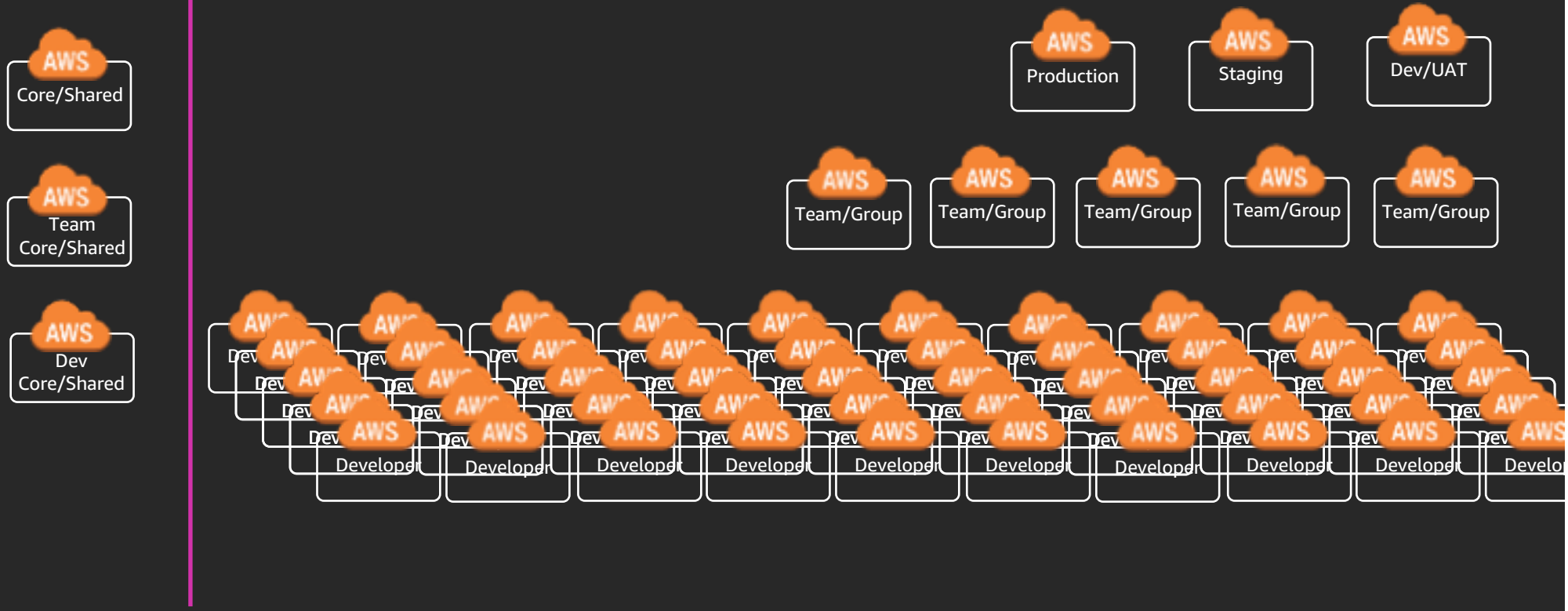
Network

Security

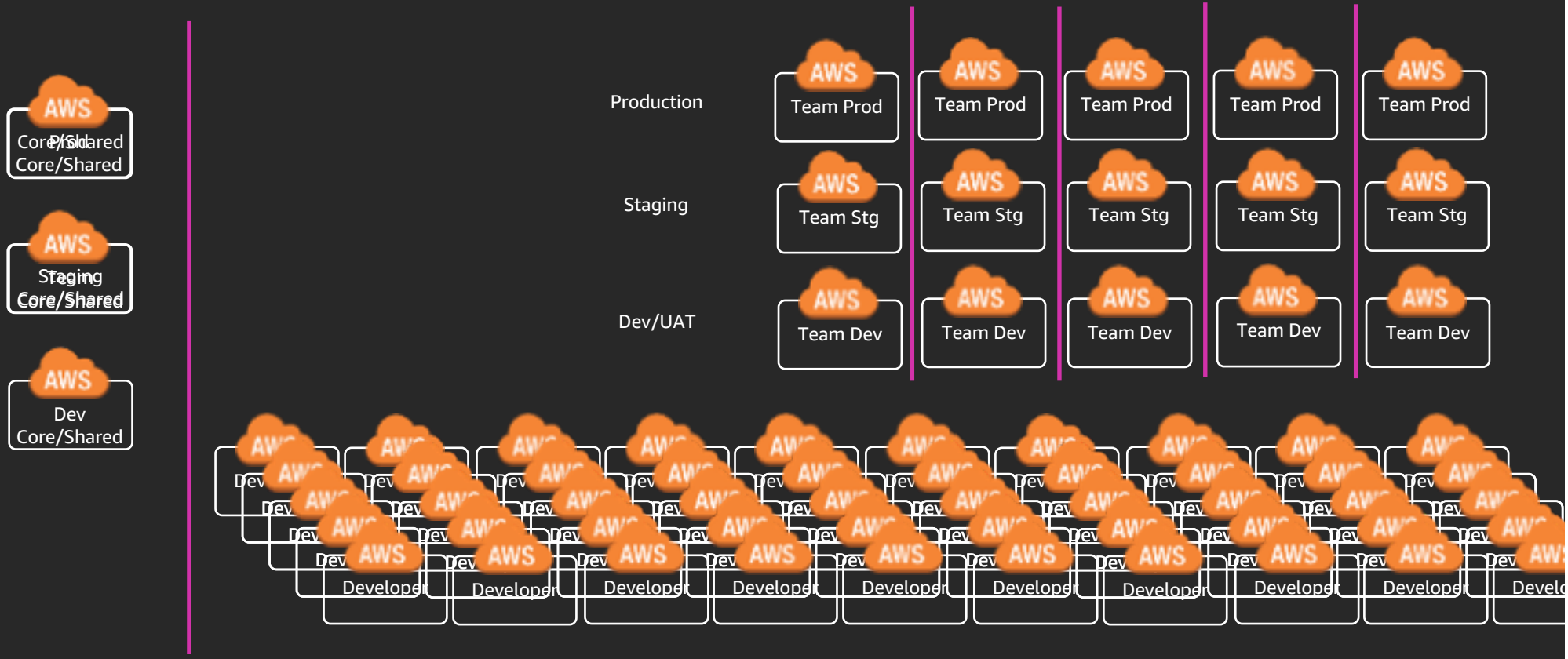
Shared by tier



Shared by tier



A different approach



AWS Account

Security/Resource Boundary

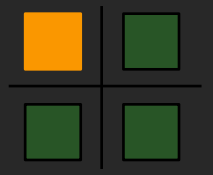
API Limits/Throttling

Billing Separation

Why one account isn't enough

Many Teams

Billing



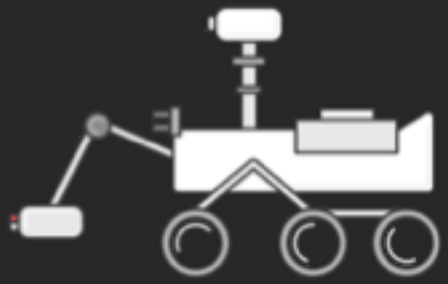
Isolation

Security / Compliance Controls

Business Process

aws

Goals



Automated

Scalable

Self-service

Guardrails NOT Blockers

Auditable

Flexible

aws

Account security considerations

Baseline Requirements

Lock

AWS Account Credential Management ("Root Account")

Enable

AWS CloudTrail
Amazon GuardDuty

Define

Map enterprise roles and permissions

Federate

Use identity solutions

Establish

InfoSec cross-account roles

Identify

Actions and conditions to enforce governance

aws

What accounts should I create?

Organizations Account

Log Archive

Security

Shared Services

Network

Billing

Sandbox

Dev

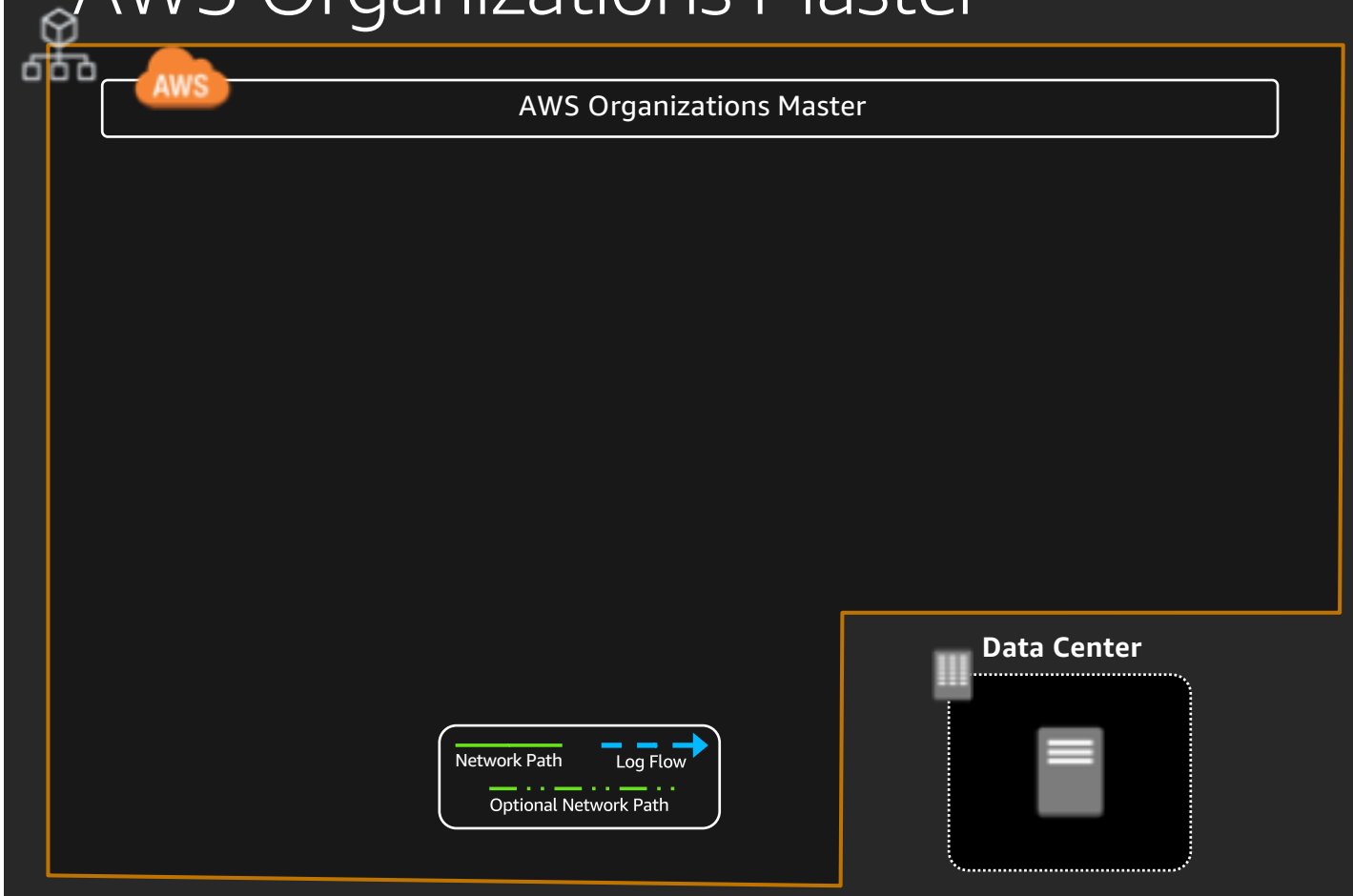
Pre-Prod

Prod

Other

aws

AWS Organizations Master



No connection to DC

Service control policies

Consolidated billing

Volume discount

Minimal resources

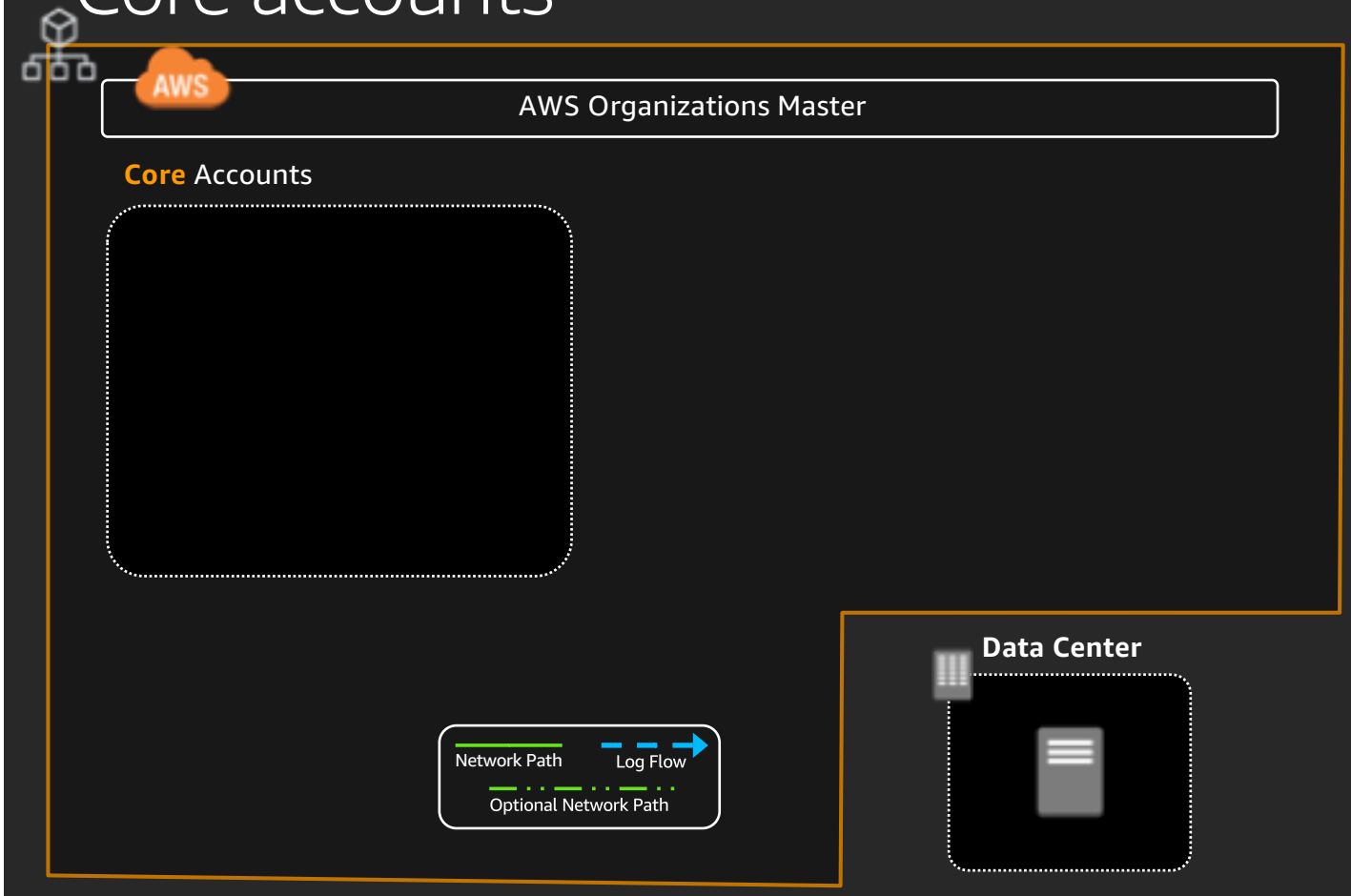
Limited access

Restrict Orgs role!

SCP: Stop CloudTrail from being disabled

```
{
  "Version": "2012-10-17",
  "Statement": [
      {
      "Effect": "Deny",
      "Action": ”cloudtrail:StopLogging",
      "Resource": "*"
    }
  ]
}
```

Core accounts



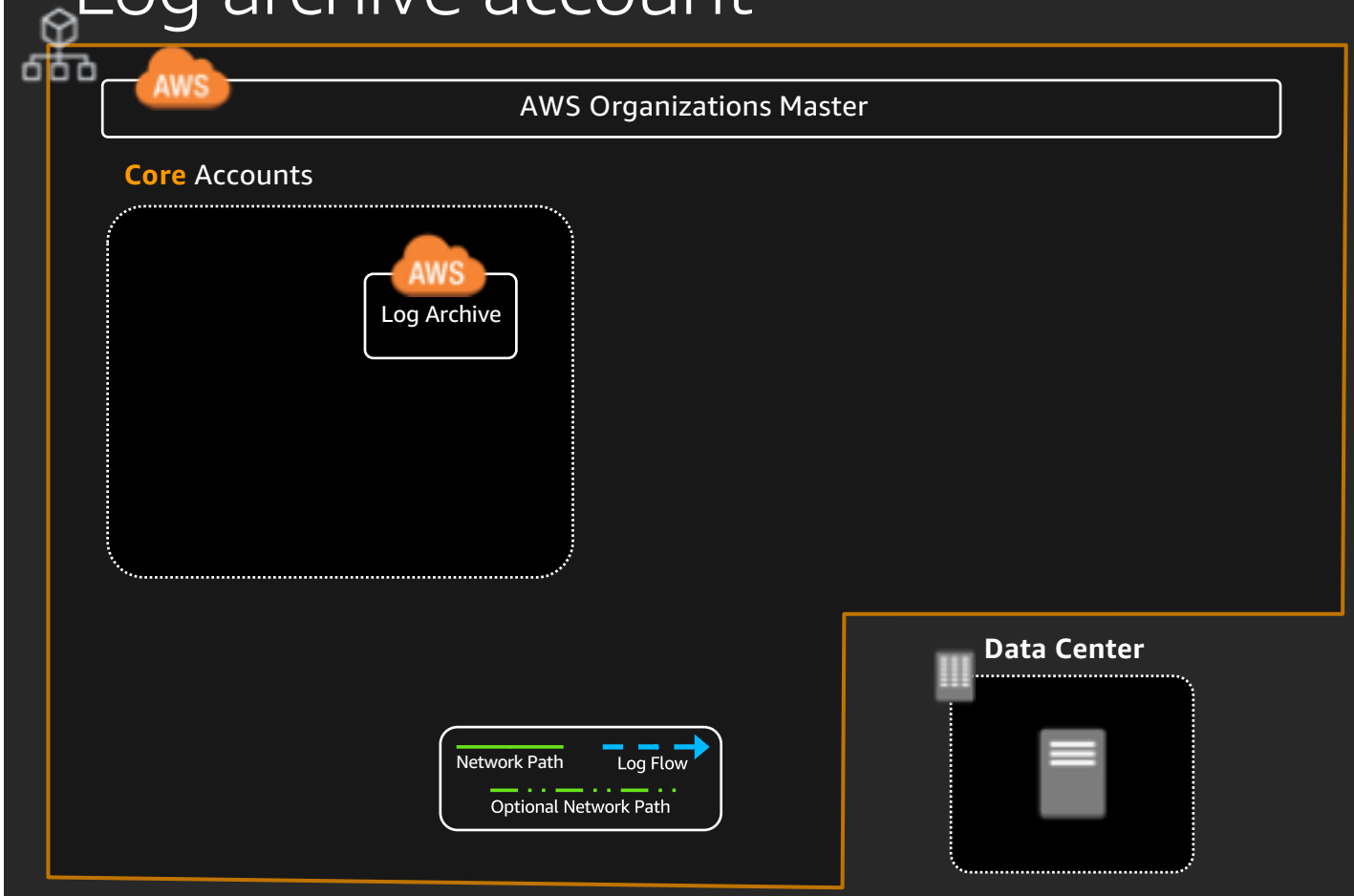
Foundational

Building Blocks

Once per organization

Have their own development life cycle (dev/qa/prod)

Log archive account



Versioned Amazon S3 bucket
Restricted
MFA delete

CloudTrail logs

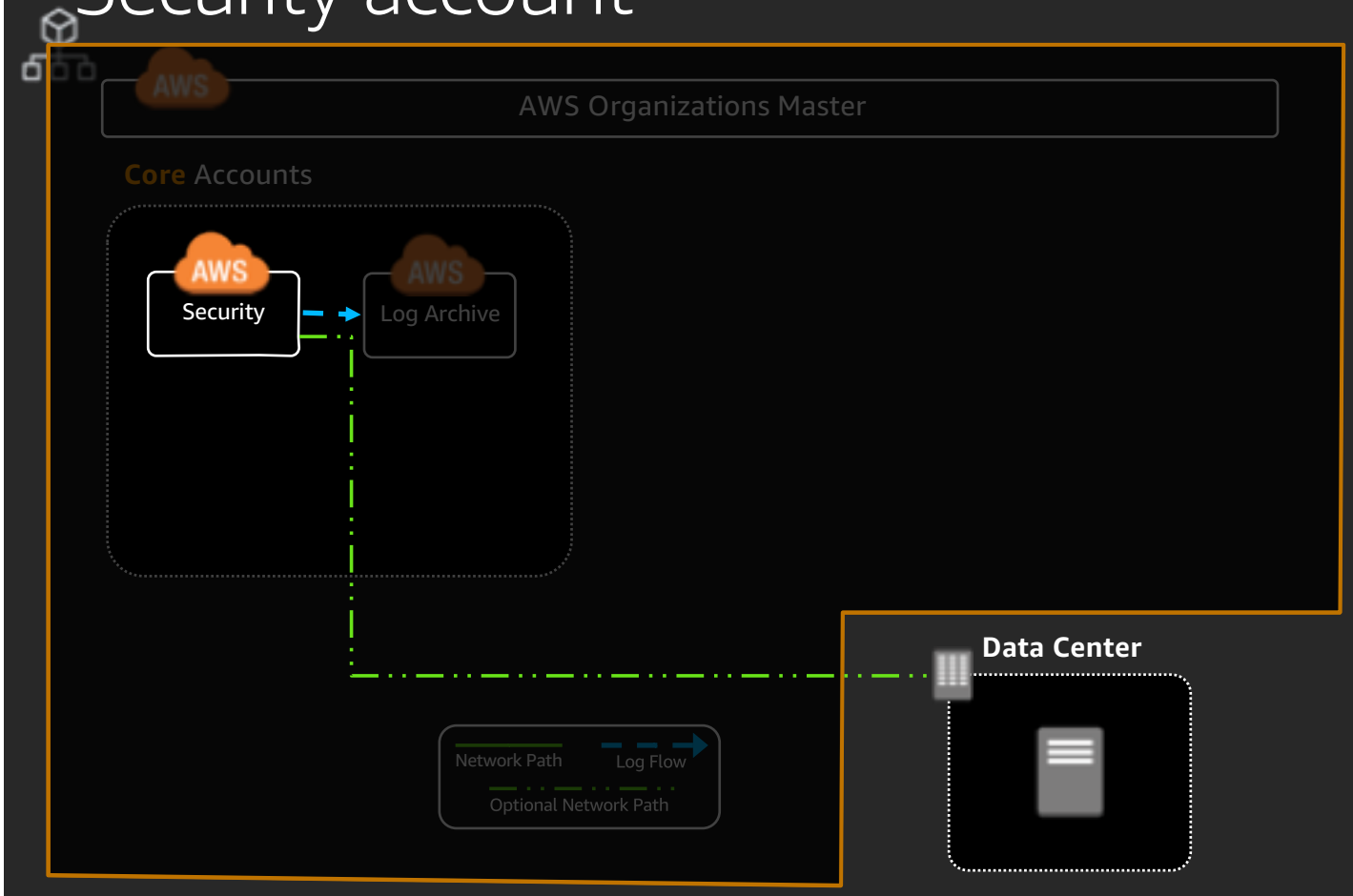
Security logs

Single source of truth

Alarm on user login

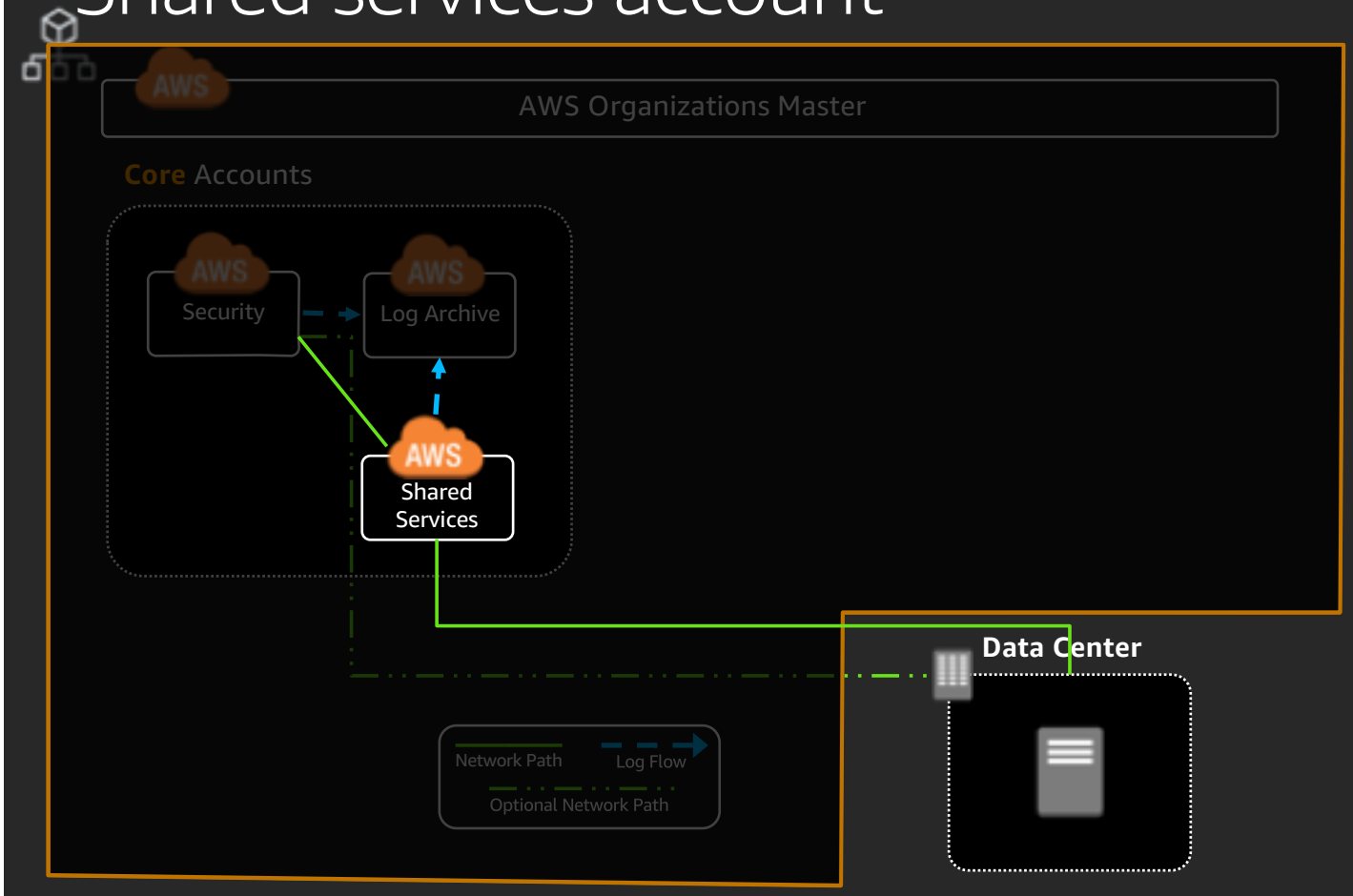
Limited access

Security account



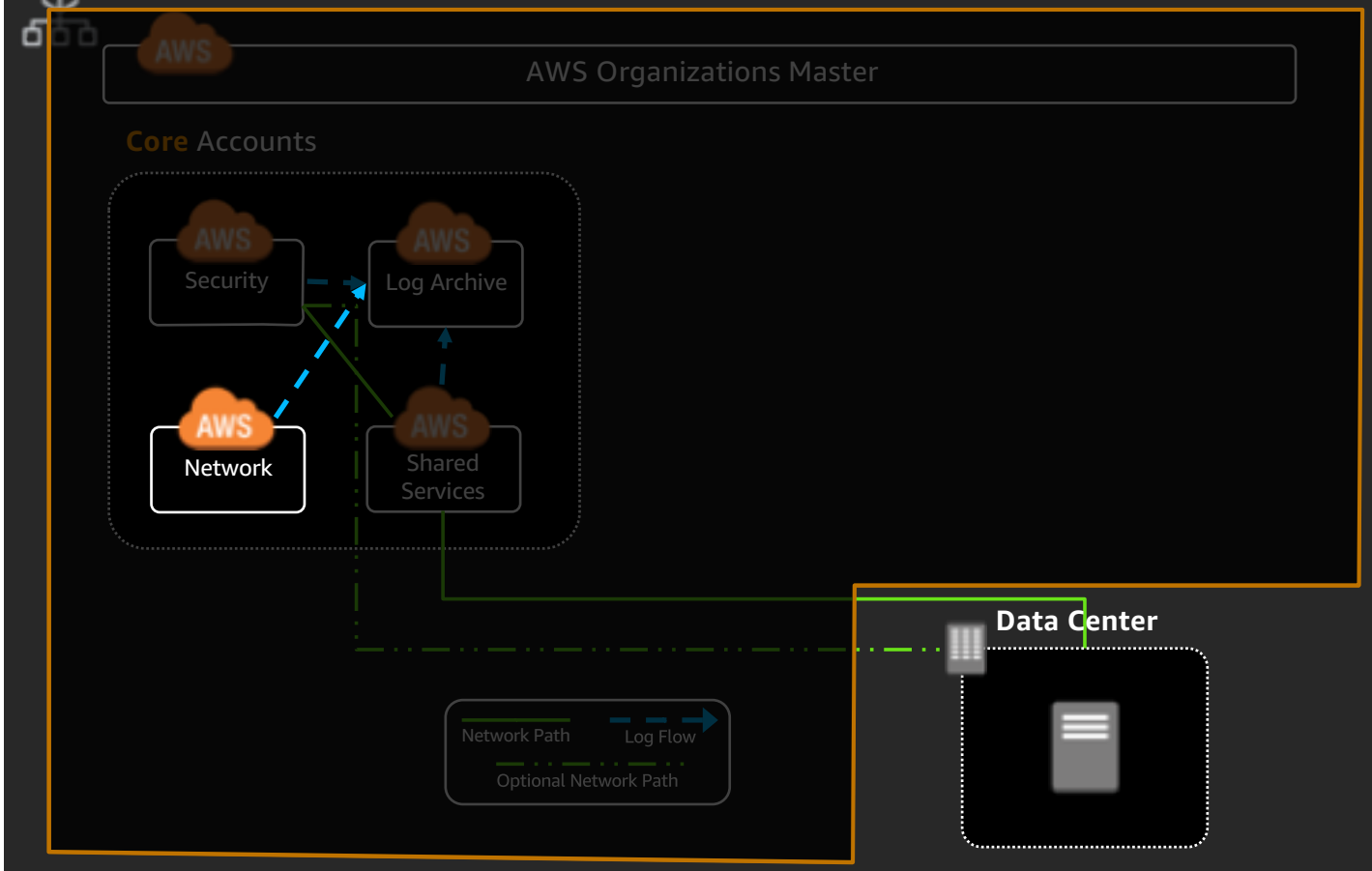
- Optional data center connectivity
- Security tools and audit
- GuardDuty Master
- Cross-account read/write
  - Automated Tooling
- Limited access

Shared services account



- Connected to DC
- DNS
- LDAP/Active Directory
- Shared Services VPC
- Deployment tools
  - Golden AMI
  - Pipeline
- Scanning infrastructure
  - Inactive instances
  - Improper tags
  - Snapshot lifecycle
- Monitoring
- Limited access

Network account



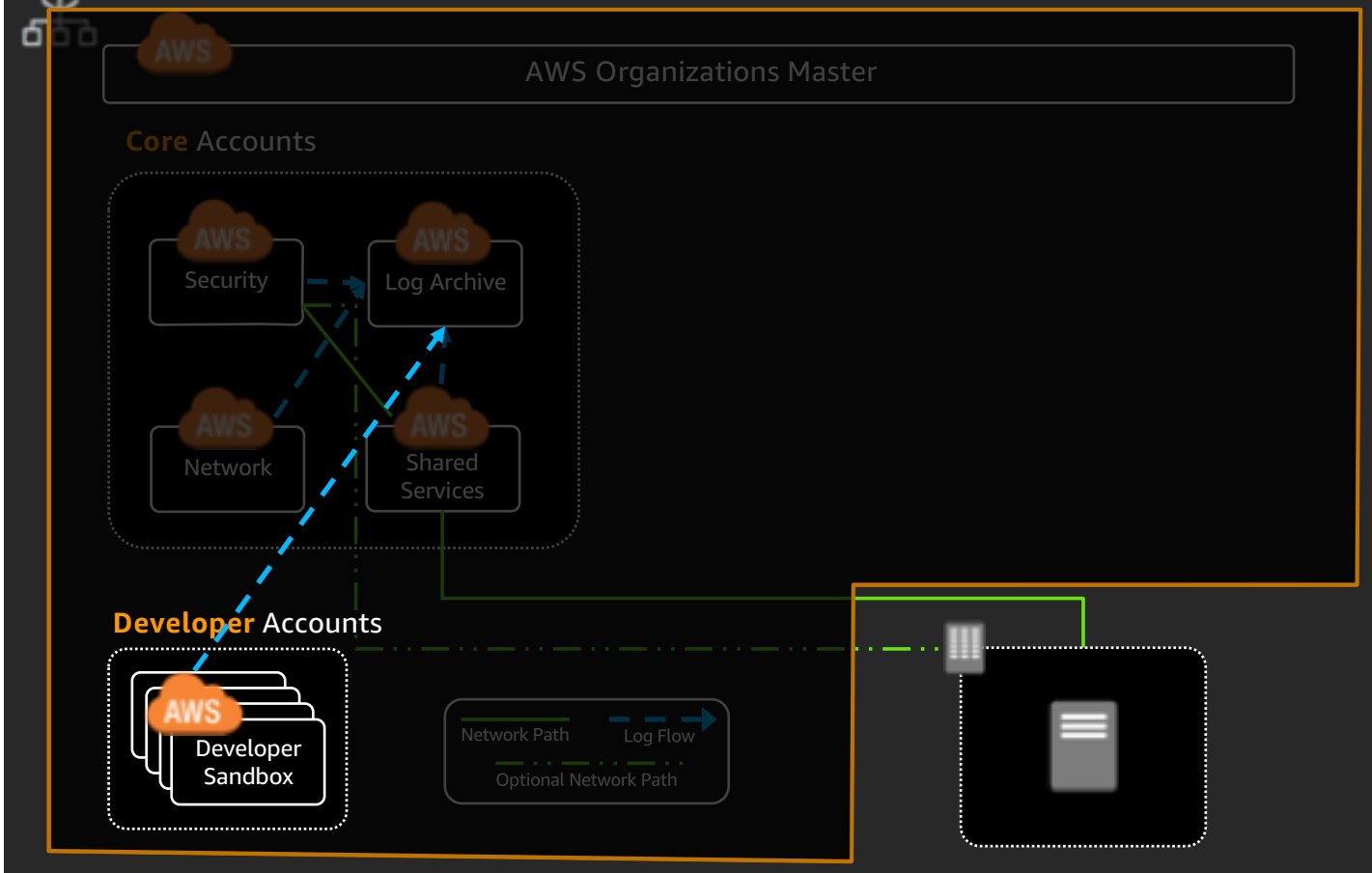
Managed by network team

Networking services

AWS Direct Connect

Limited access

Developer sandbox



No connection to DC

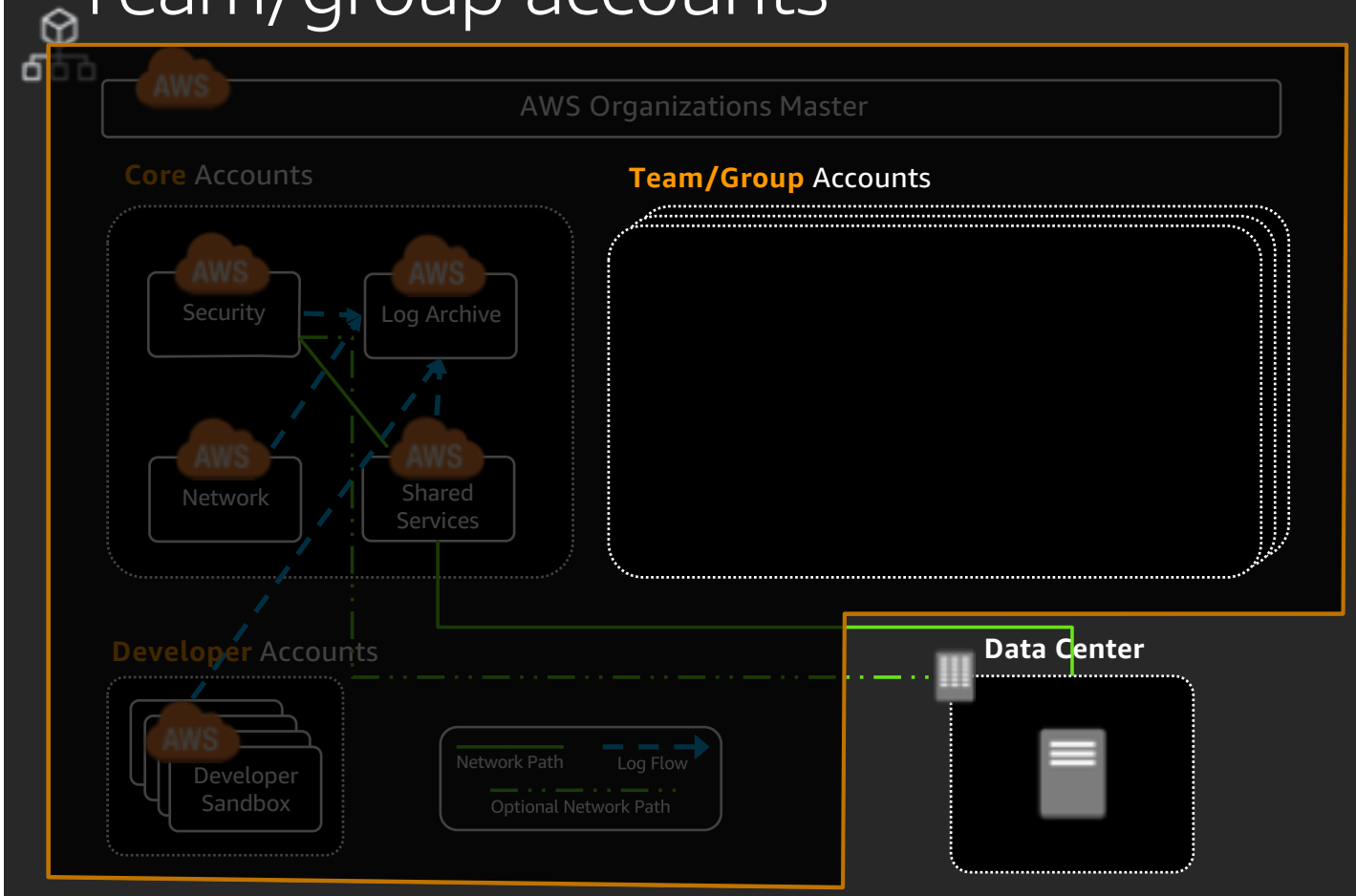
Innovation space

Fixed spending limit

Autonomous

Experimentation

Team/group accounts

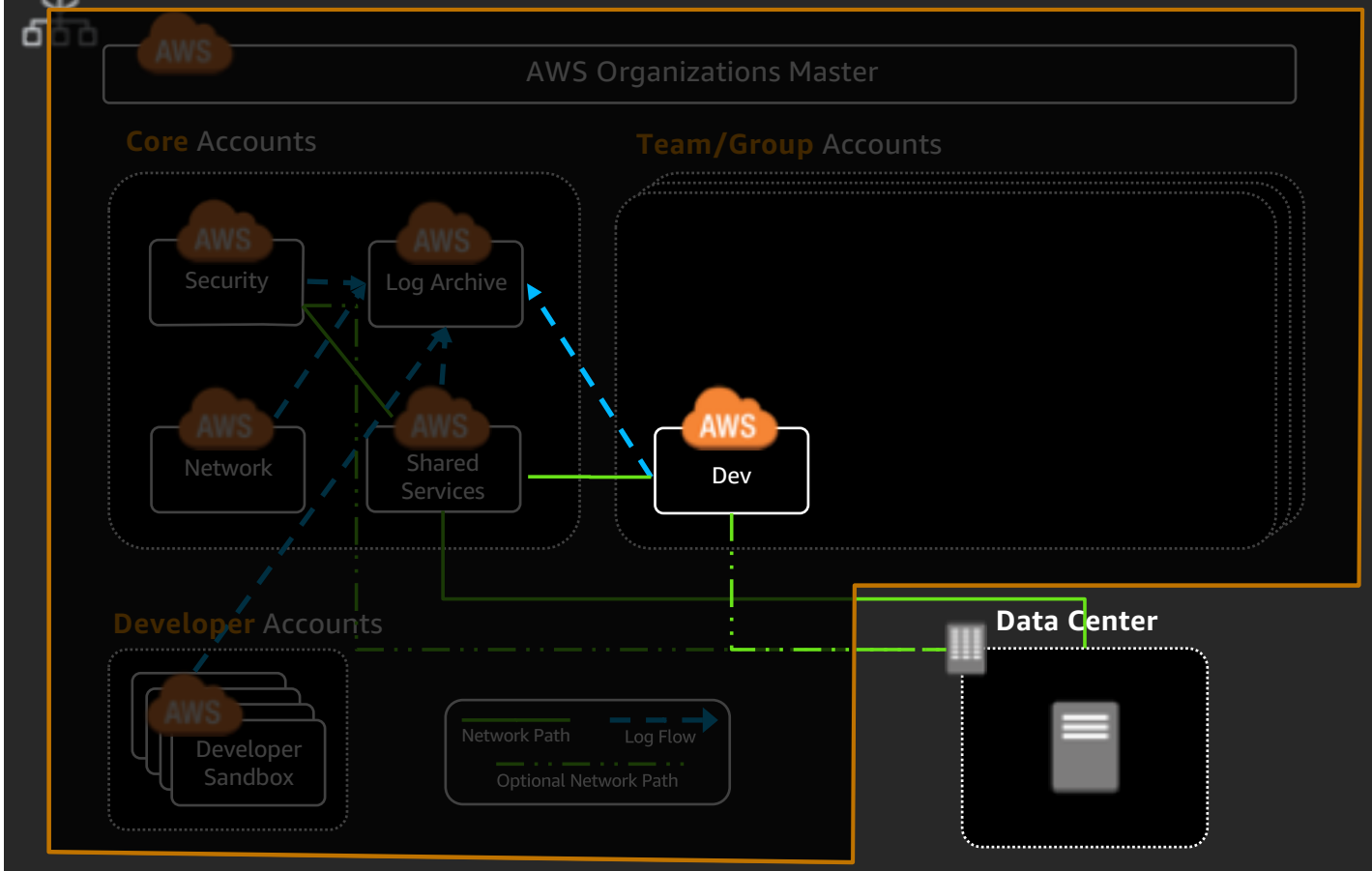


Based on level of needed isolation

Match your development lifecycle

Think Small

Dev

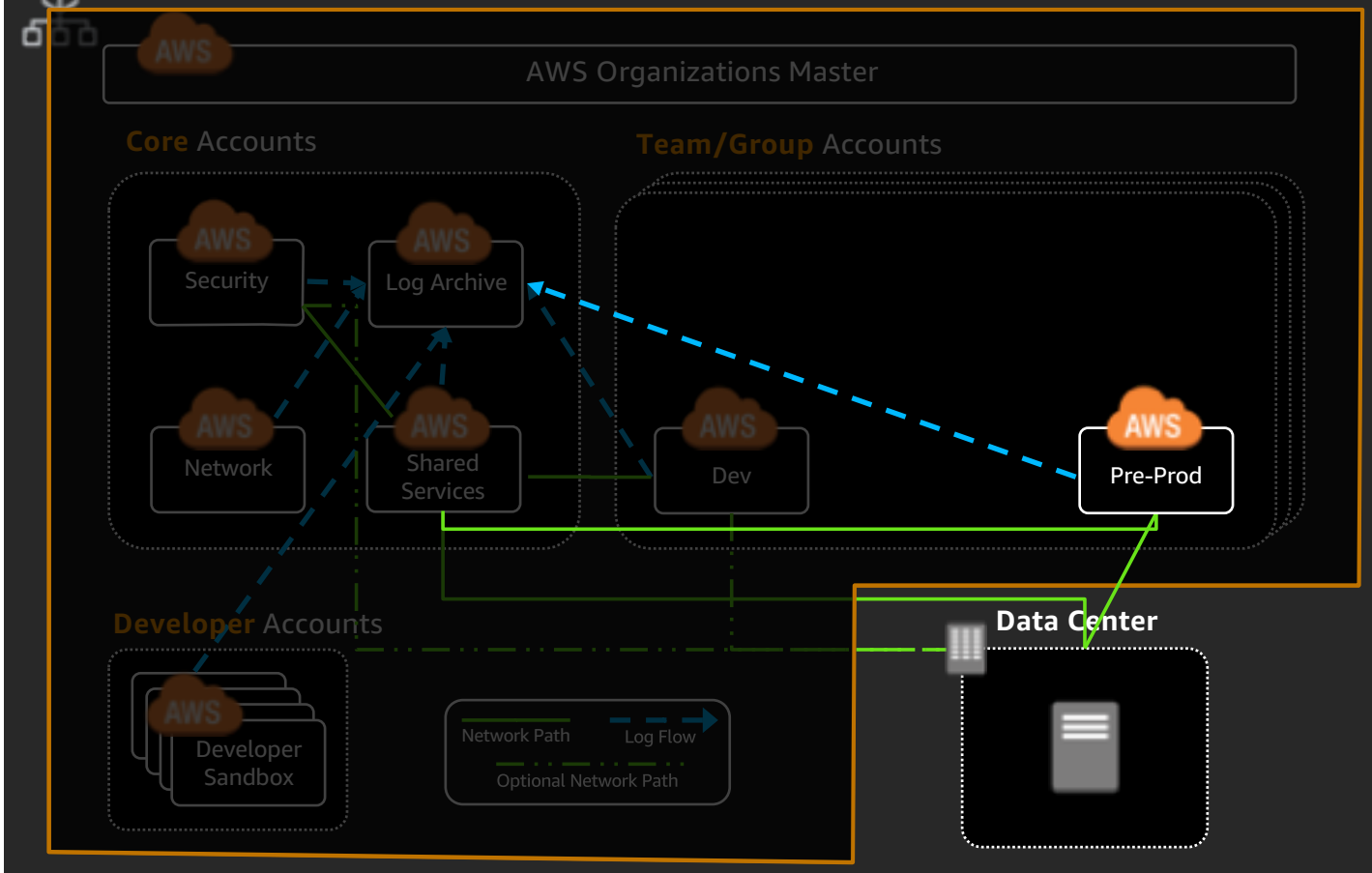


Develop and iterate quickly

Collaboration space

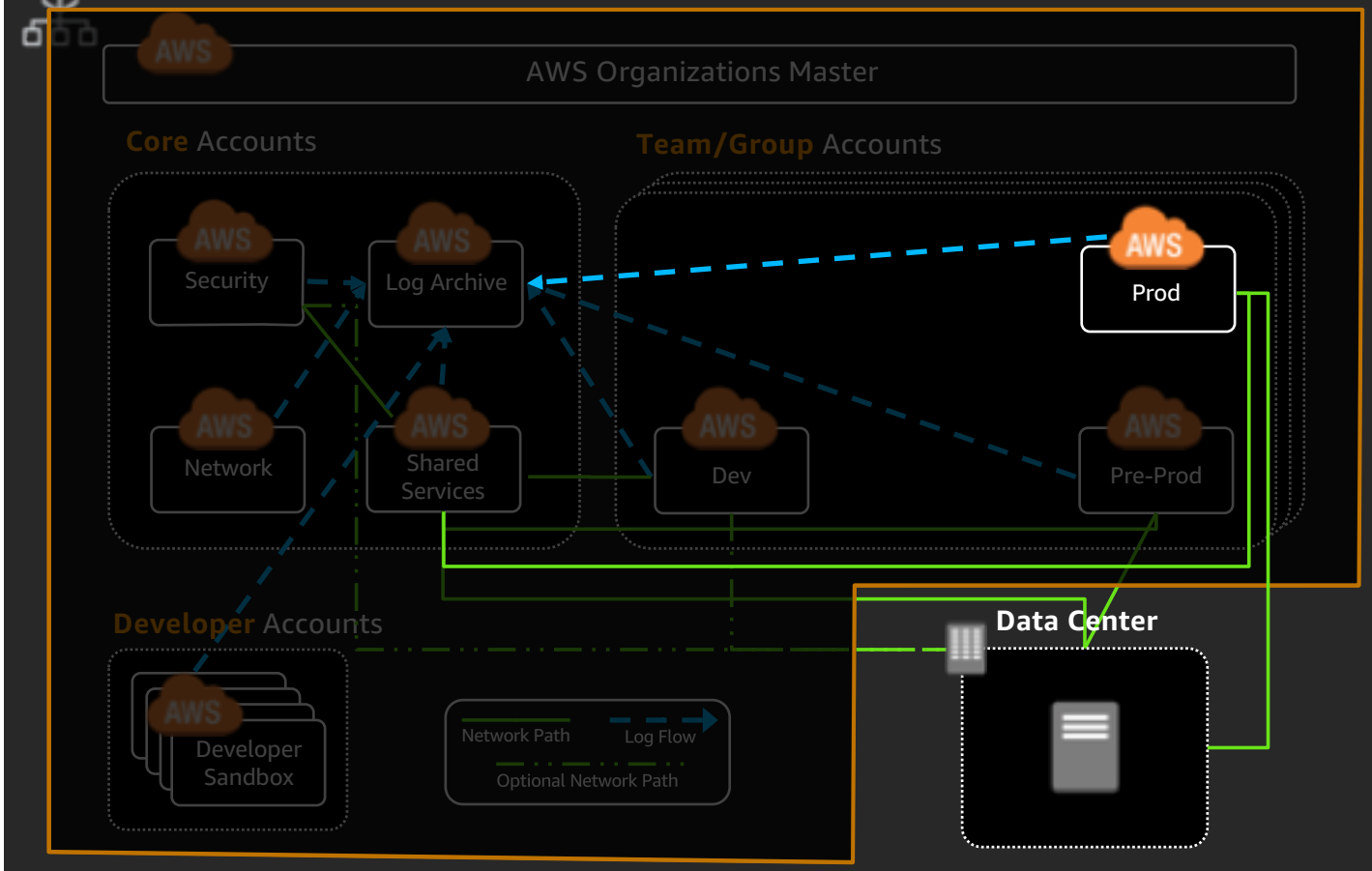
Stage of SDLC

Pre-production



- Connected to DC
- Production-like
- Staging
- Testing
- Automated Deployment

Production



Connected to DC

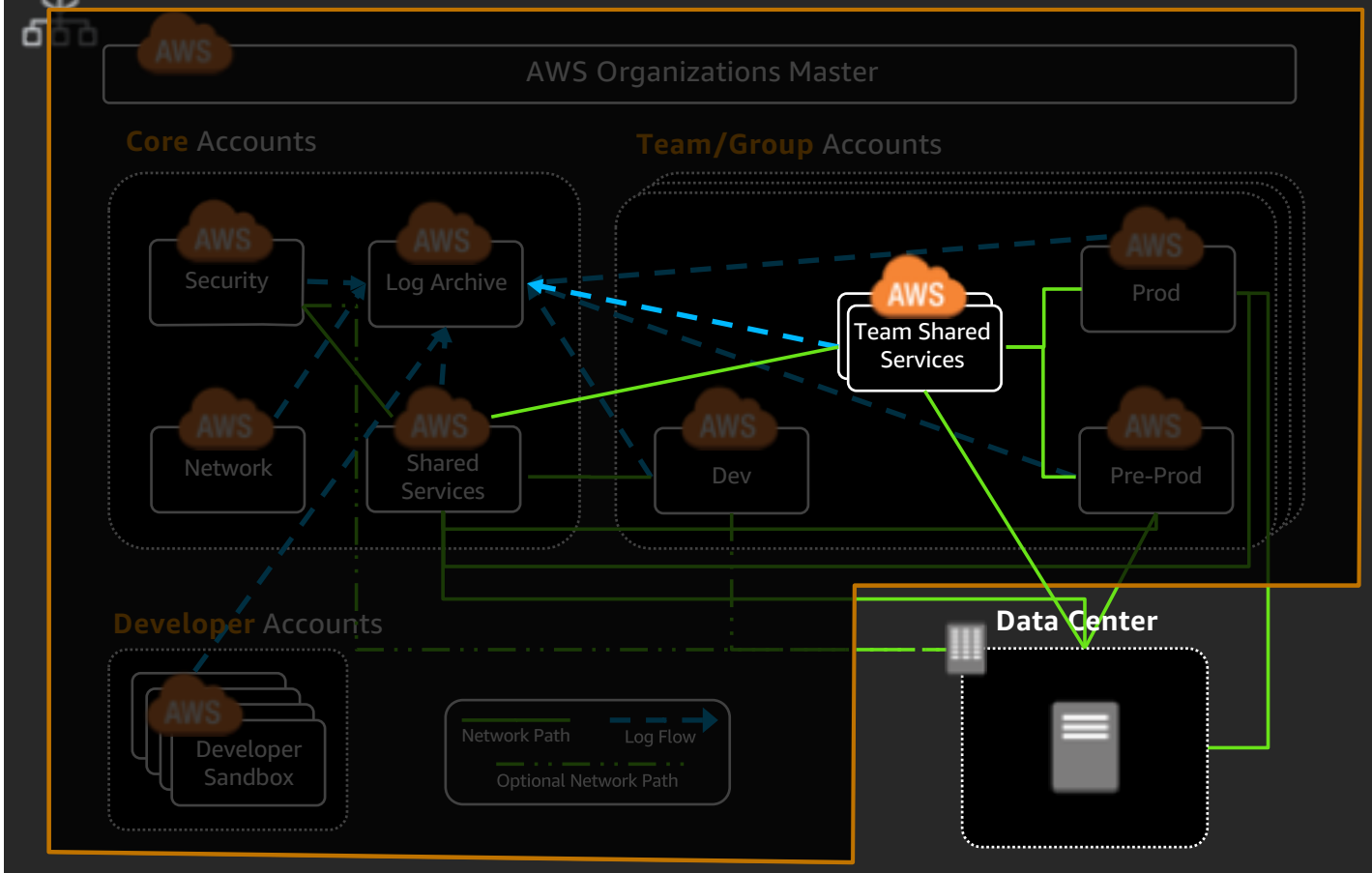
Production applications

Promoted from Pre-Prod

Limited access

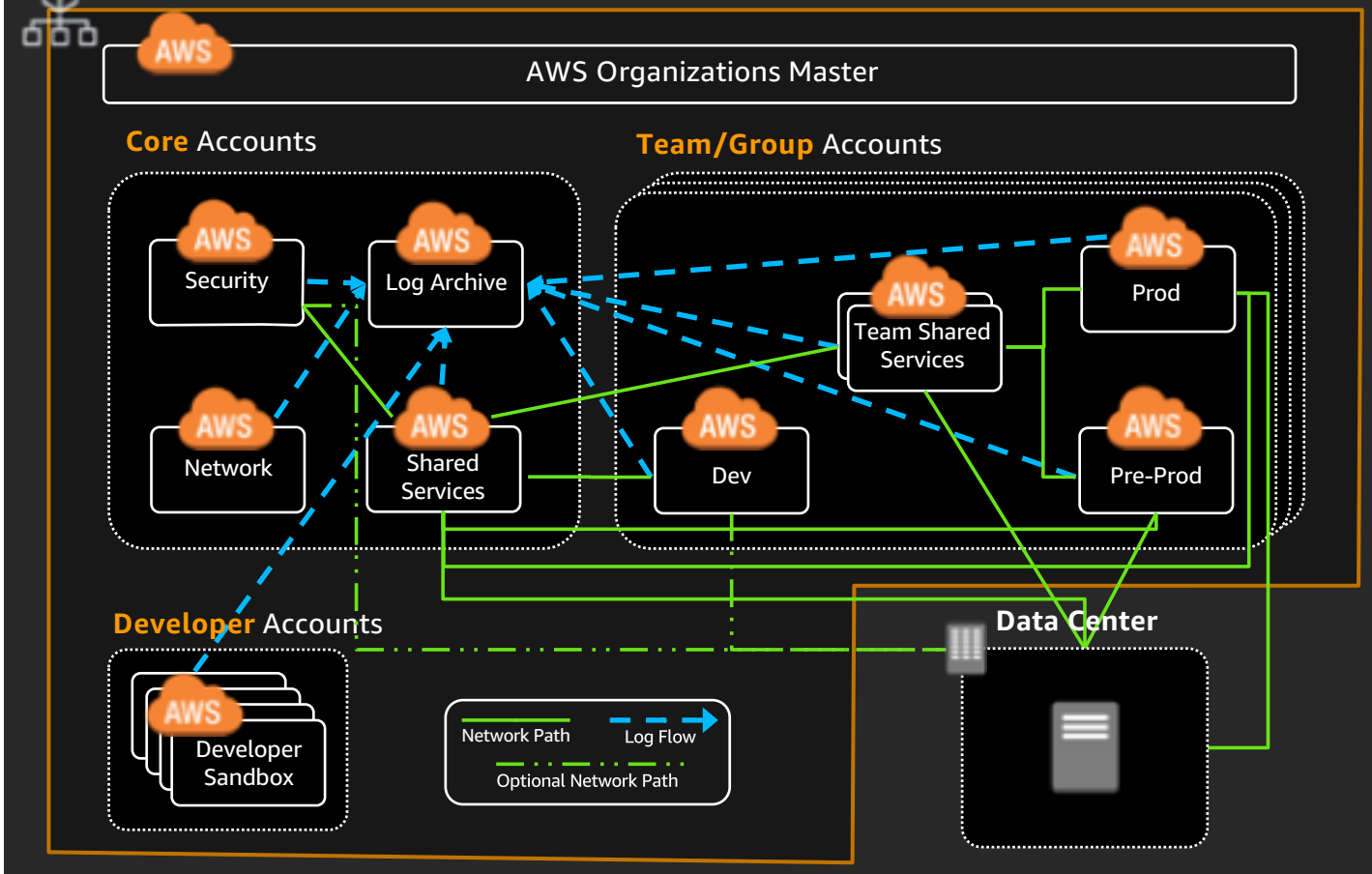
Automated Deployments

Team shared services



- Grows organically
- Shared to the team
- Product-specific common services
- Data lake
- Common tooling
- Common services

Multi-account approach



Orgs: Account management

Log Archive: Security logs

Security: Security tools, AWS Config rules

Shared services: Directory, limit monitoring

Network: Direct Connect

Dev Sandbox: Experiments, Learning

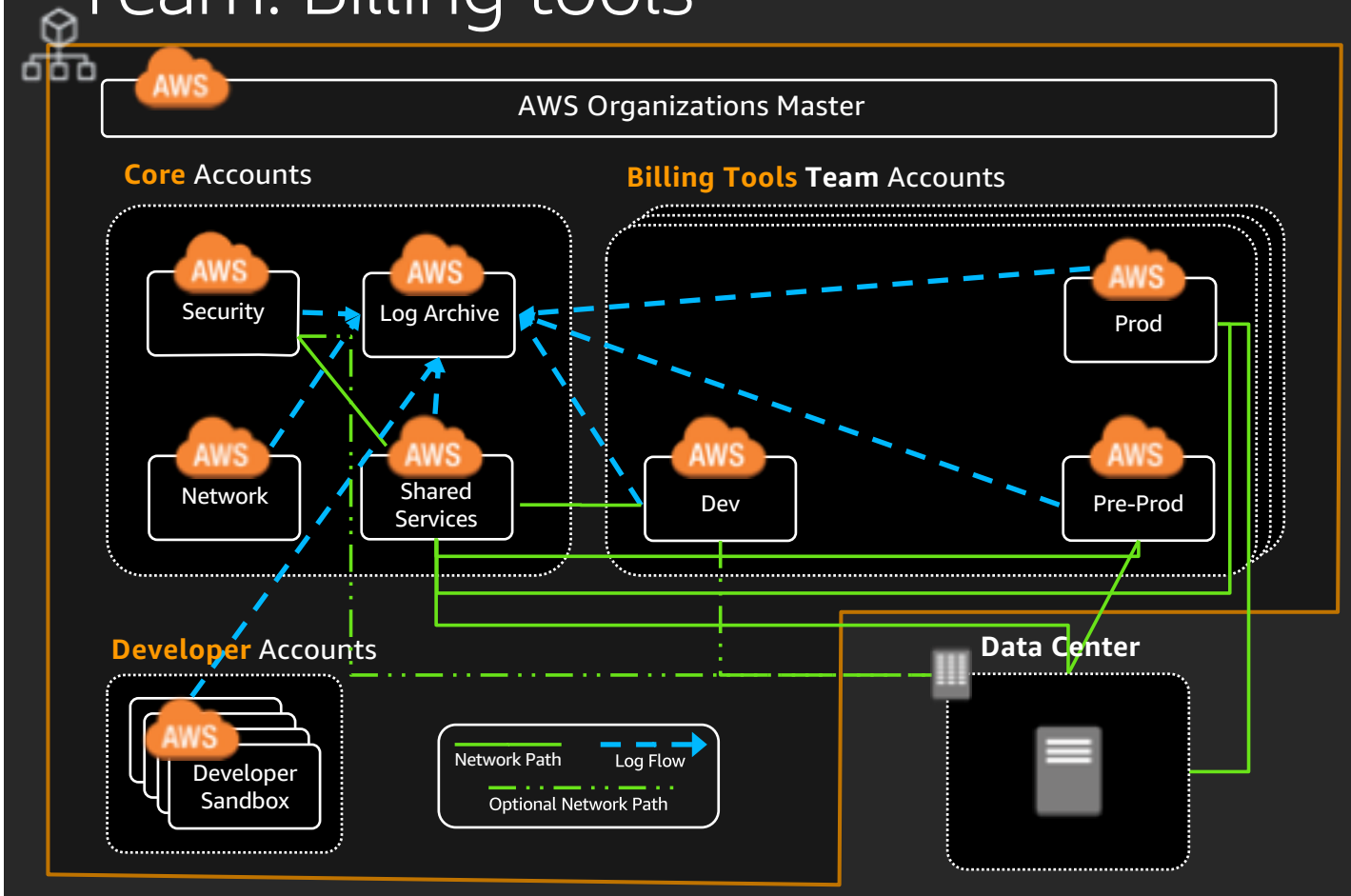
Dev: Development

Pre-Prod: Staging

Prod: Production

Team SS: Team Shared Services, Data Lake

Team: Billing tools



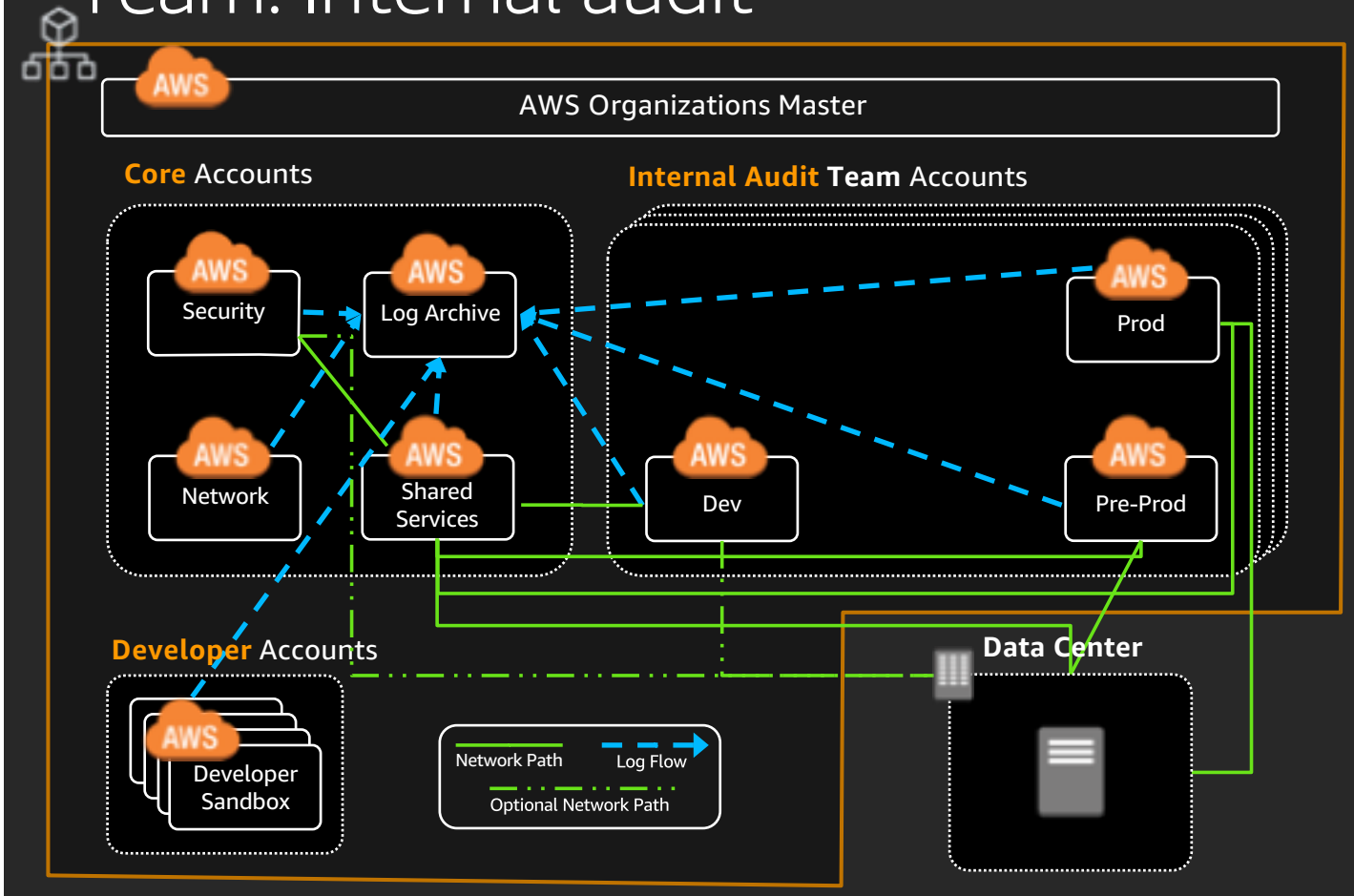
Reduces access to Organizations account

Billing reports

Usage metrics and reporting

Usage optimizations and RI management

Team: Internal audit

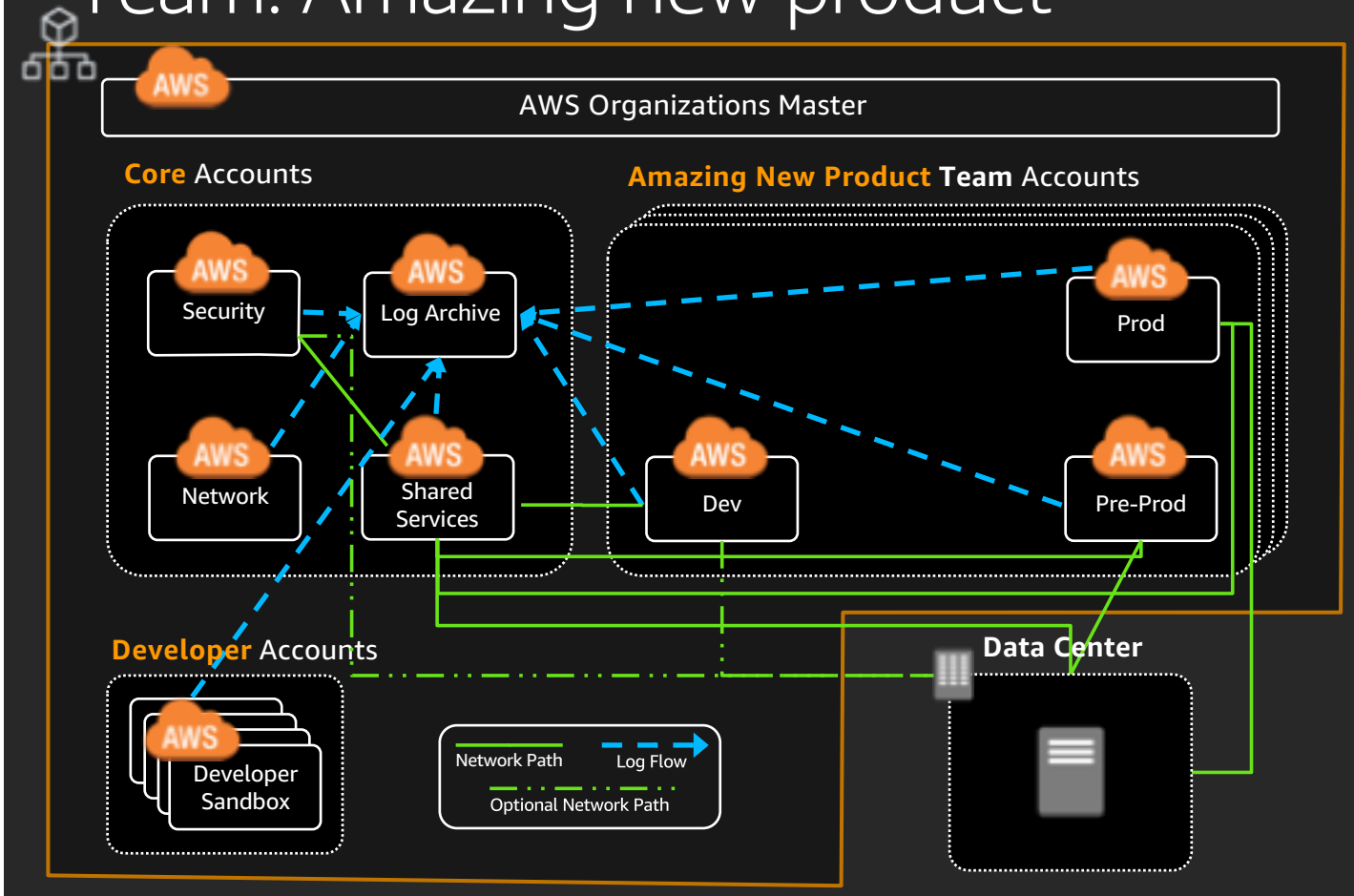


Regulatory compliance

Read-only access to needed logs

Limited access

Team: Amazing new product

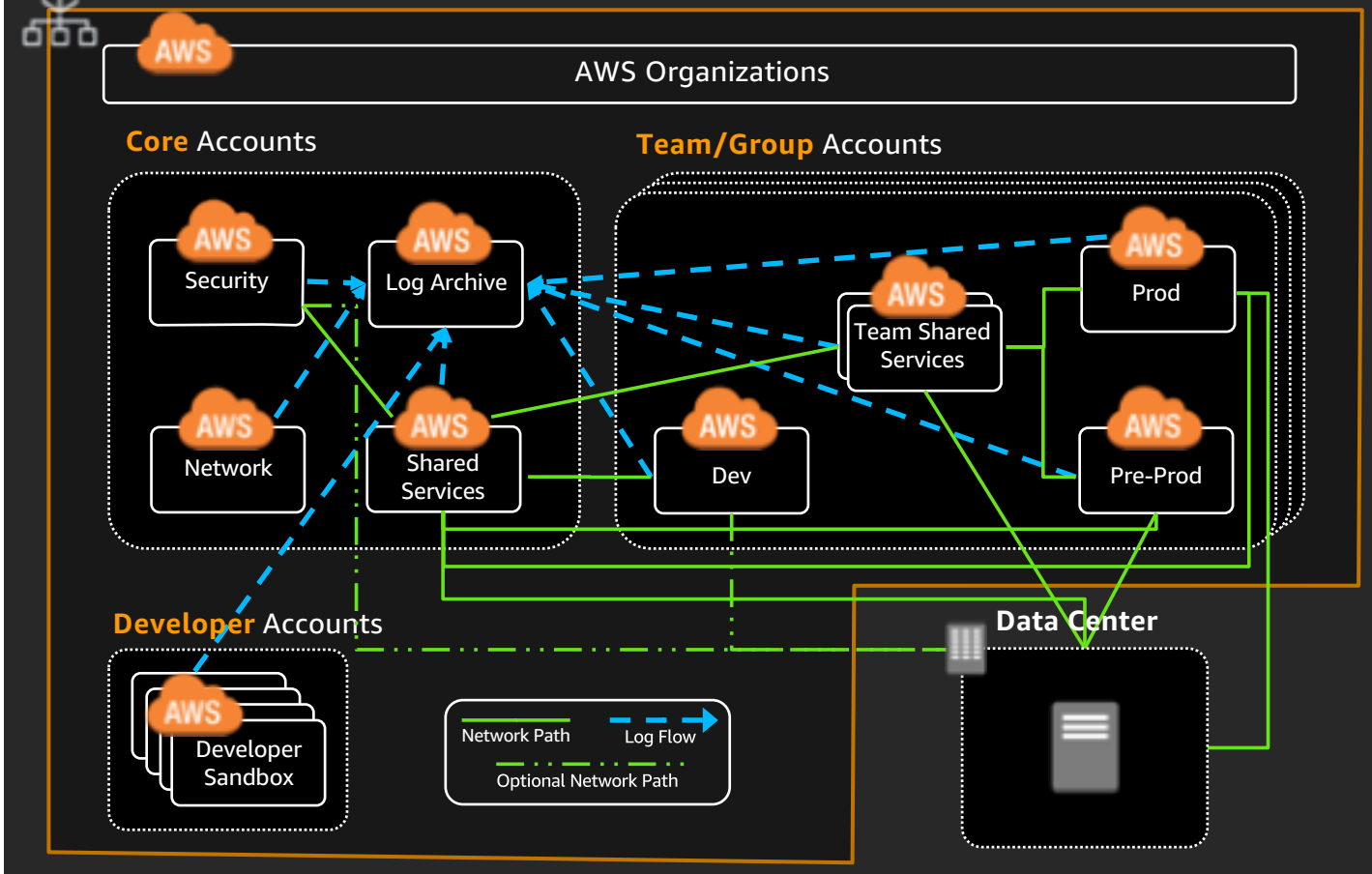


Match your development lifecycle

Think Small

Summary

Multi-account approach



Orgs: Account management

Log Archive: Security logs

Security: Security tools, AWS Config rules

Shared services: Directory, limit monitoring

Network: Direct Connect

Dev Sandbox: Experiments, Learning

Dev: Development

Pre-Prod: Staging

Prod: Production

Team SS: Team Shared Services, Data Lake

Next steps

Define tagging strategy

Define automation strategy

Create Organizations Master account

Create Log Archive account

Create Security account

Create Shared Services account

Create Developer Sandbox account(s)

Action plan

Create Organizations Master account
- Create temporary s3 bucket for CloudTrail logs
- Enable CloudTrail locally
- Enable organizations full feature

Create Log Archive account
- Create bucket(s) for security logs (CloudTrail, AWS Config)
  - Enable MFA delete
  - Enable versioning
  - Define limited access bucket policy
  - Add SCP to prevent s3:delete
- Backfill: Enable CloudTrail in organizations master account to send logs to Log Archive account
- Backfill: Copy CloudTrail logs for actions that happened between Organizations Master creation and log archive

Create Security account
- Backfill: Cross-account roles with trust to security account for organizations master and log archive
- Read-only role
- Read/Write role (fewer permissions for assumption)
- <CommonCheckList>
- Create security tooling/Lambda functions for security checks

Create Shared Services account
- <CommonCheckList>
- Connect via DX/VPN to DC
- Launch common services
  - Directory services
  - Limit monitoring

Create AWS Network account
- Order your Direct Connect
- <CommonCheckList>

Common checklist

- Secure Root credentials
  MFA
  - OTP
  - U2F could make this easier for managing them
    - https://aws.amazon.com/blogs/security/how-to-create-and-manage-users-within-aws-sso/
- Complex password
  - Establish rotation policy
  - Link to Organizations Master account if not already a member
  - Use group email/phone as the contact info
  - Enable CloudTrail in all regions, send to Log Archive account
  - Enable GuardDuty in all regions.
  - Security Account as GuardDuty master
    - Operationalize the findings
- Enable AWS Config, send to Log Archive account
- Enable appropriate AWS Config rules
  - s3 bucket encryptions
  - s3 world read/write
  - ebs encryption etc...
- Create read-only cross-account Security role
- Create read/write cross-account Security role
- Create VPC (non-overlapping IP space)
- Enable federation into account
  - http://federationworkshopreinvent2016.s3-website-us-east-1.amazonaws.com/
- Define roles and access policies
- Peer/Privatelink VPC with Shared Services
- Add a policy for prefix naming conditions to every account—For example, deny access to Lambda functions that start with “security*”
- Review CIS Foundations Benchmark and leverage as appropriate

Next steps

Define tagging strategy

Define automation strategy

Create Organizations Master account

~~Create Log Archive account~~

~~Create Security account~~

~~Create Shared Services account~~

Create Developer Sandbox account(s)

Action Plan

Create Organizations Master account
- Create temporary s3 bucket for CloudTrail logs
- Enable CloudTrail locally
- Enable organizations full feature

~~**Create Log Archive account**~~
- ~~Create bucket(s) for security logs (CloudTrail, AWS Config)~~
  - Enable MFA delete
  - Enable versioning
  - Define limited access bucket policy
  - Add SCP to prevent s3:delete
- Backfill: Enable CloudTrail in organizations master account to send logs to Log Archive account
- Backfill: Copy CloudTrail logs for actions that happened between Organizations Master creation and log archive

~~**Create Security account**~~
- Backfill: cross-account roles with trust to security account for organizations master and log archive
- ~~Read-only role~~
- ~~Read/Write role (fewer permissions for assumption)~~
- <CommonCheckList>
- Create security tooling/Lambda functions for security checks

~~**Create Shared Services account**~~
- <CommonCheckList>
- Connect via DX/VPN to DC
- Launch common services
  - ~~Directory services~~
  - Limit monitoring

Create AWS Network account
- Order your Direct Connect
- <CommonCheckList>

Common Checklist

- Secure Root credentials
  MFA
  - OTP
  - U2F could make this easier for managing them
    - https://aws.amazon.com/blogs/security/how-to-create-and-manage-users-within-aws-sso/
- Complex password
  - Establish rotation policy
  - ~~Link to Organizations Master account if not already a member~~
  - Use group email/phone as the contact info
  - ~~Enable CloudTrail in all regions, send to Log Archive account~~
  - ~~Enable GuardDuty in all regions.~~
  - ~~Security Account as GuardDuty master~~
    - Operationalize the findings
- Enable AWS Config, send to Log Archive account
- ~~Enable appropriate AWS Config rules~~
  - ~~s3 bucket encryptions~~
  - ~~s3 world read/write~~
  - ~~ebs encryption etc…~~
- ~~Create read-only cross-account Security role~~
- ~~Create read/write cross-account Security role~~
- Create VPC (non-overlapping IP space)
- ~~Enable federation into account~~
  - http://federationworkshopreinvent2016.s3-website-us-east-1.amazonaws.com/
- Define roles and access policies
- Peer/Privatelink VPC with Shared Services
- Add a policy for prefix naming conditions to every account—For example, deny access to Lambda functions that start with “security*”
- Review CIS Foundations Benchmark and leverage as appropriate

aws

Thank you!