



AppGate SDP Introduction

Basic architecture and key concepts

Cyxtera

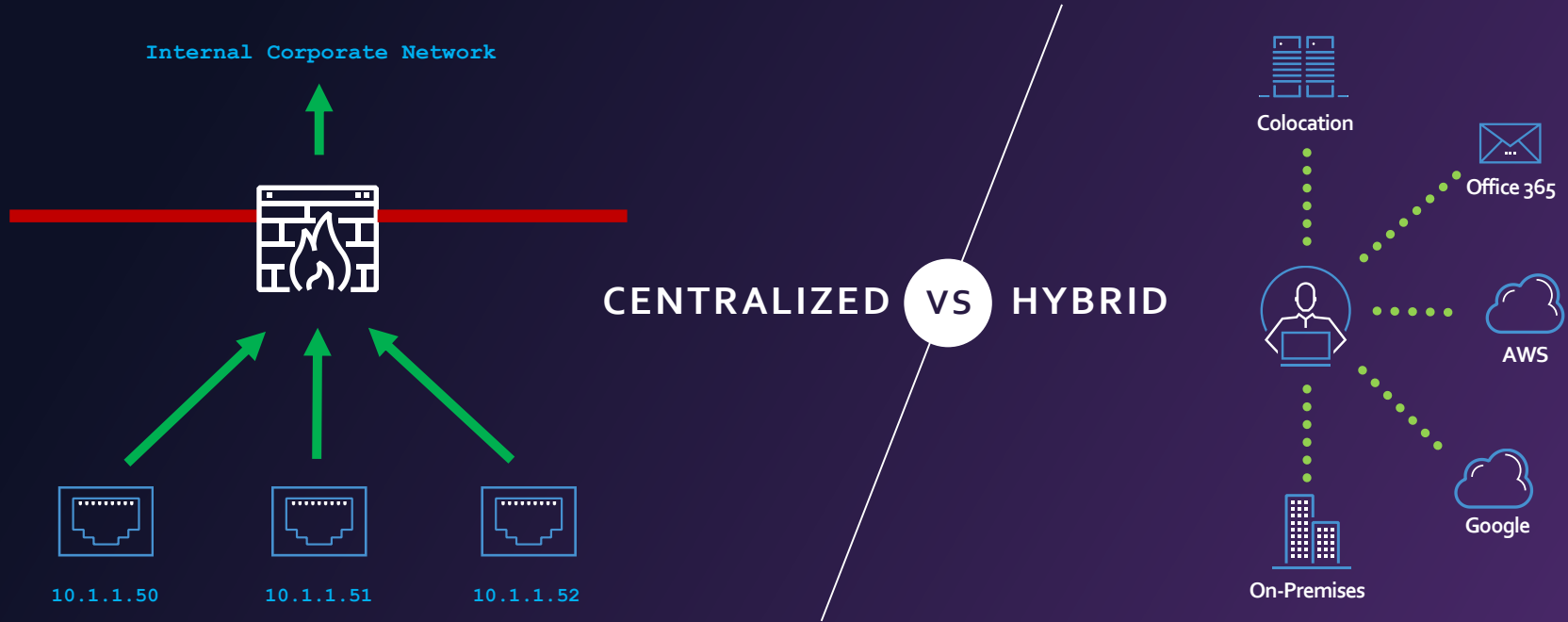
Yesterday's network security doesn't work for modern I.T.

1996

2019

Perimeter security has remained largely unchanged for the past 2 decades.
I.T. has moved on – distributed, dynamic and interconnected.

Hybrid I.T. has killed the perimeter.



Firewall rules are binary and static.



Should this IP block have access to this network (Y/N)?

But today's business isn't.

Is Jim's machine patched?

What's the current security posture?

Where is he?

What time is it?

What project is Jim working on?



What are his credentials?

Should Jim have access to the production SAP database server?

A better approach to network security:

Software-Defined Perimeter



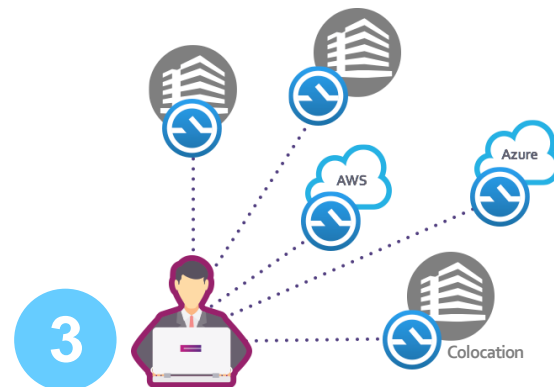
Identity-centric

- User- or device- based access control
- Integrates with directory services and IAM
- Context sensitive



Zero-trust model

- Authentication before connection
- Dynamically-provisioned 1:1 connectivity
- Unauthorized resources completely dark



Built like cloud, for cloud

- Distributed, stateless and highly scalable
- Programmable and adaptive
- Dynamic and on demand

AppGate creates a "Segment of One"

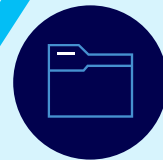
Fine-grained controls reduce attack surface



1:1 ENCRYPTED NETWORK SEGMENT

Dynamic, 1:1 network segment

- Encrypted for greater security
- User sees only their authorized resources
- All other resources completely dark, inaccessible
- Entitlements adjusted in real time as necessary



PROTECTED RESOURCES
Cloud, Hybrid or On-Premises

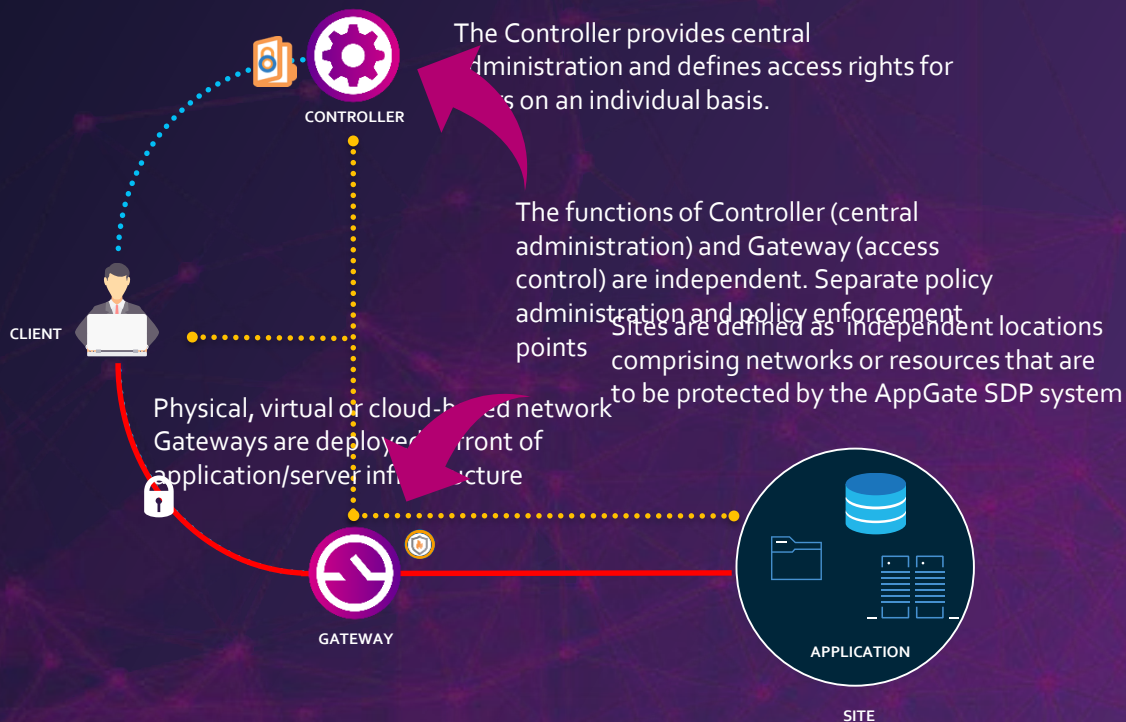


AppGate SDP Components

- **Client**
 - Runs silently, runs everywhere: win/mac/linux/android/ios
- **Controller**
 - Authentication, Authorization, Policy Management, Identity Provider Integrations etc.
- **Gateway**
 - VPN Server, Firewall, Real-time enforcer
- **LogServer (optional)**
 - Store, parse, query, visualize audit logs

How AppGate works

- 1 Client makes access request to the Controller (the Policy decision point). Typically linked with Identity Providers. Contains the Policies that define Entitlements (Access context).
- 2 Controller checks context, authenticates and authorizes user verifying any number of claims within each session and passes the Entitlements token to the Client.
- 3 The Client passes the token on to the Gateways which provision a micro-firewall instance started on a separate thread just for that user. The Gateway then translates the Entitlements into a set of individualized micro-firewall rules.
- 4 A Segment of One network is built for this session. For each packet received from the Client, the correct rules allow, conditionally allow or block access. AppGate continuously monitors for any context changes, adapts user access accordingly.
- 5





AppGate SDP 8 key concepts

Cyxtera

AppGate SDP – The 8 key concepts

Zero-trust defense

Hyper-scaling

Robust, easy to deploy architecture

Highly available – always on connectivity

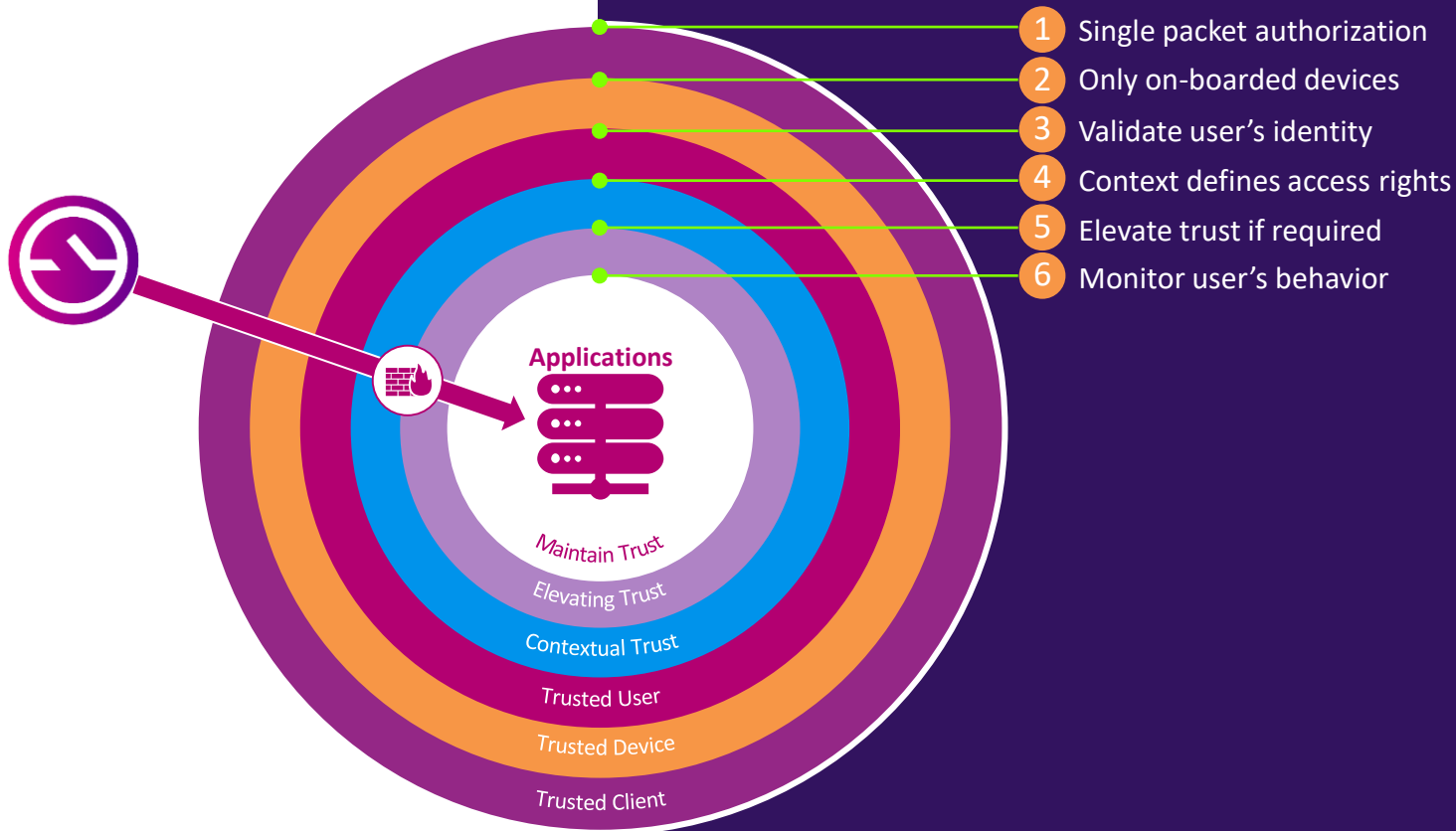
Designed around the user

Dynamic resource resolving

Adaptive authorization

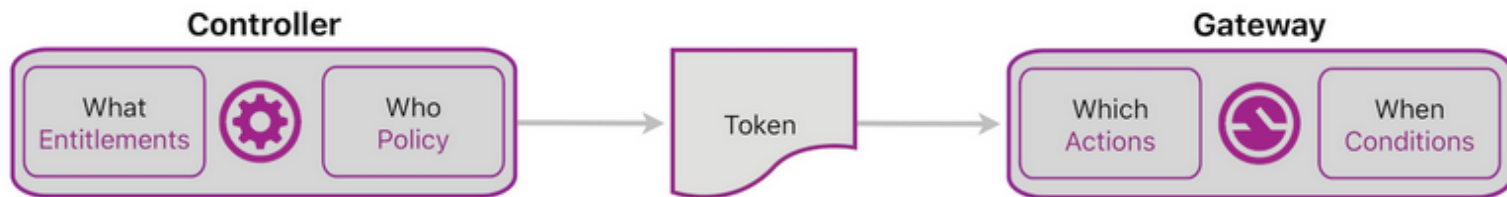
Role based API driven administration

Zero-trust defense



Robust, easy to deploy architecture

Tokens are used to pass information between the Controller, Client and Gateway. They contain all the information needed for authentication, authorization and real-time access control.



Appliance based – physical, virtual or cloud instances

Controllers - stateless

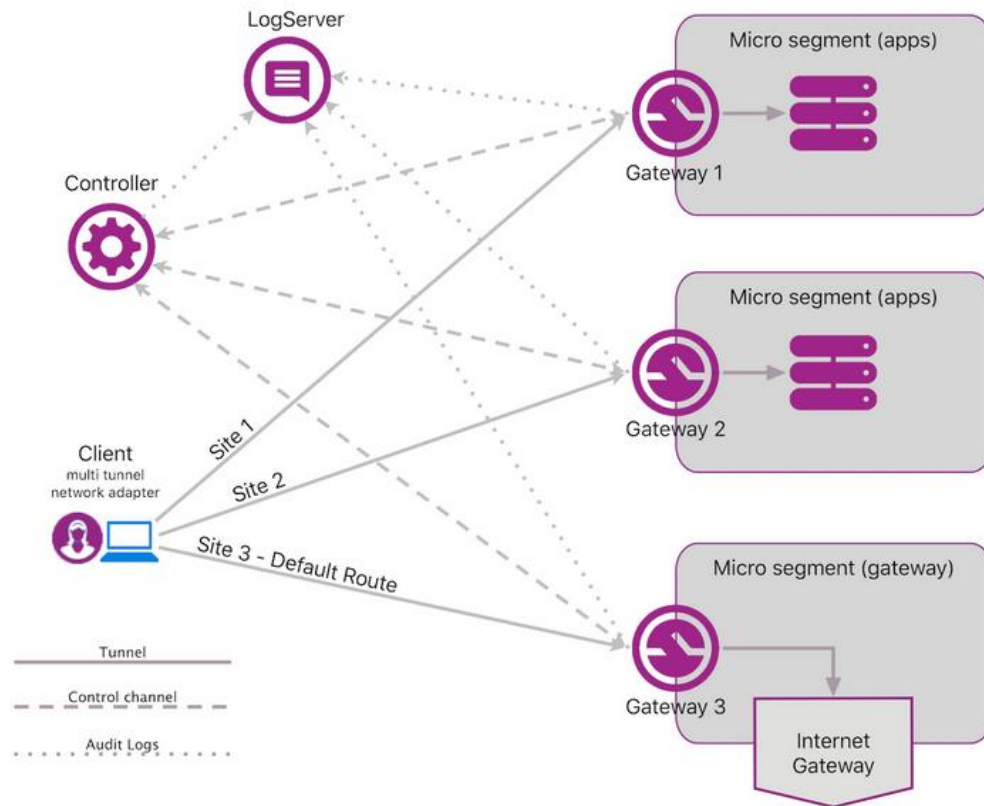
- Secure token issuing service
- Client passes-on renewable tokens
- Rebooting Controller won't affect existing tokens/users

Gateways - no pre-set firewall rules

- No real-time Controller <> Gateway communication
- No Gateway <> Gateway communication
- Runs a separate firewall service for each user

Designed around the user

- AppGate SDP works by creating direct one-to-one TLS/DTLS connections between users and the Sites they need to access – multi tunnel network adapter
- Functions of Controller (central administration) and Gateway (access control) are independent
LogServer can also be added
- Concept of Sites allows Gateways to be deployed to protect any group of target hosts, defined subnets



Adaptive authorization

Claims are key-value pairs that relate to the identity and context of the user and device. 3 types of claims:

- User: often static claims such as username from the IdP
- Device: often dynamic claims such as IP address from the connecting device
- System: often dynamic claims such as country code from the Gateway

Policy Assignment - Active when any ▾ below are true

user.username is testuser
user.username is demo

Conditional Access - Allowed when all ▾ below are true ([Switch to Editor mode](#)) [Add new](#)

Day of week is monday, tuesday, wednesday, thursday or friday

Two classes of availability within the system:

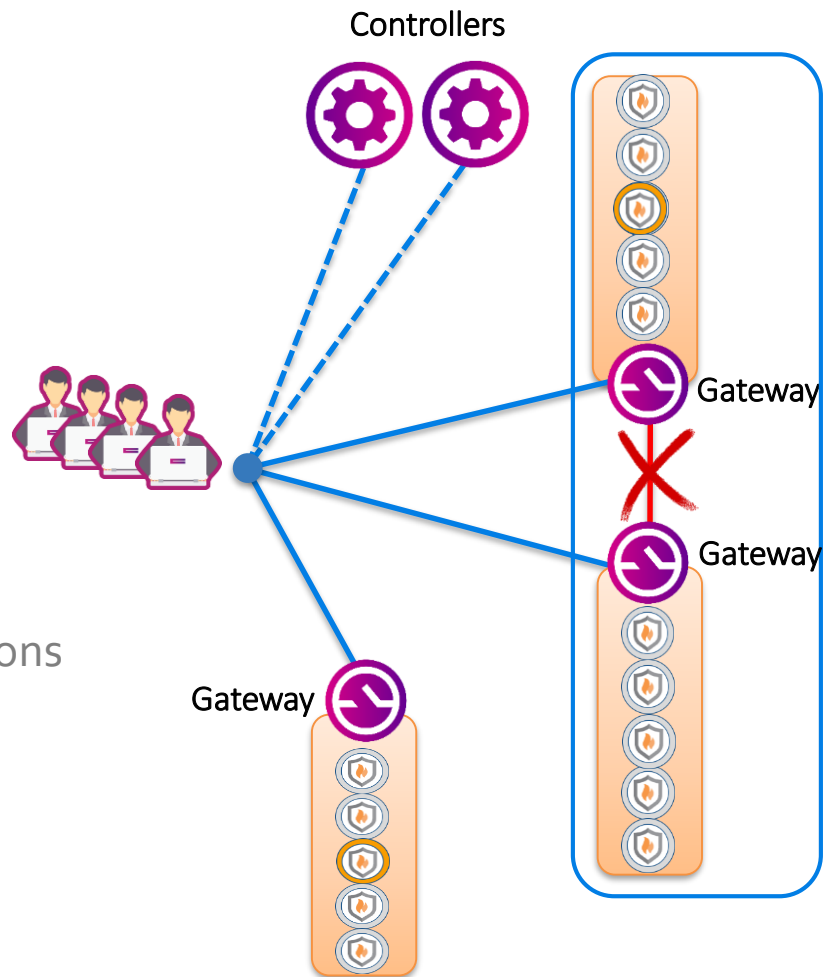
- Fixed: set by the system - will always be gathered.
- On-demand: configured by the admin

Map On-demand Device Claims [Add new](#)

Run Device Script mapped to device.version on All macOS devices
Process Running mapped to device.firefox on All platforms

Hyper-scaling

- Multi-Controller group (up to 6 currently)
- Multi-Master Controller
 - Background replication
- Support for multiple sites
- Multiple Gateways per site
- HA Gateways
- Multiple per-user firewall services
- No wasteful Gateway <> Gateway communications
- Multi-tunnel network adapter
- Autonomous clients



Highly available – always on connectivity

Client auto-connects in the background

Gateway syncs states once connected:

Tiny states – can be synced to client

States signed - tamper proof

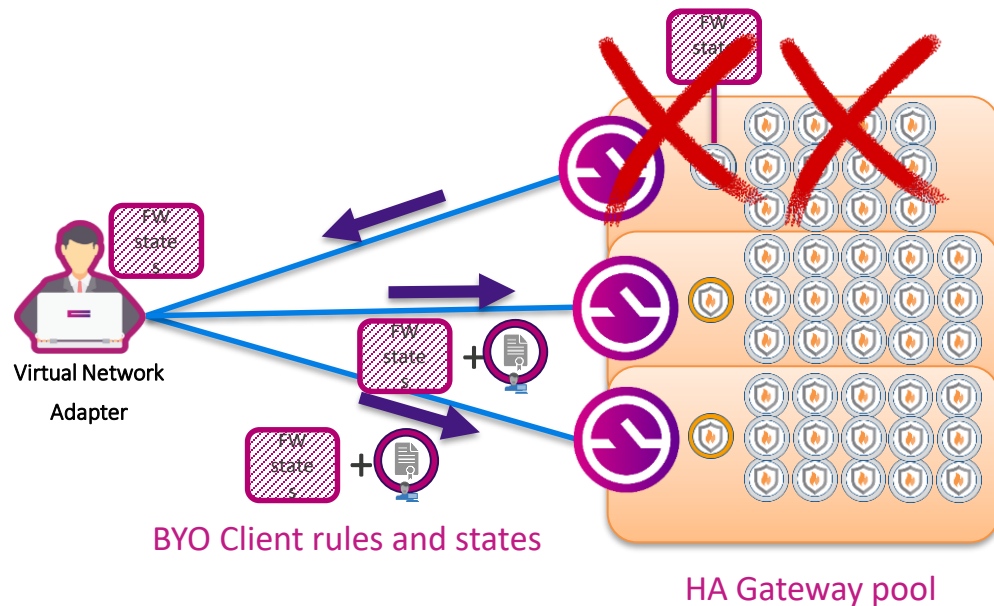
States sent - to device memory

Gateway fails:

Tokens & states transferred to any other member of Gateway pool

Same model used for roaming:

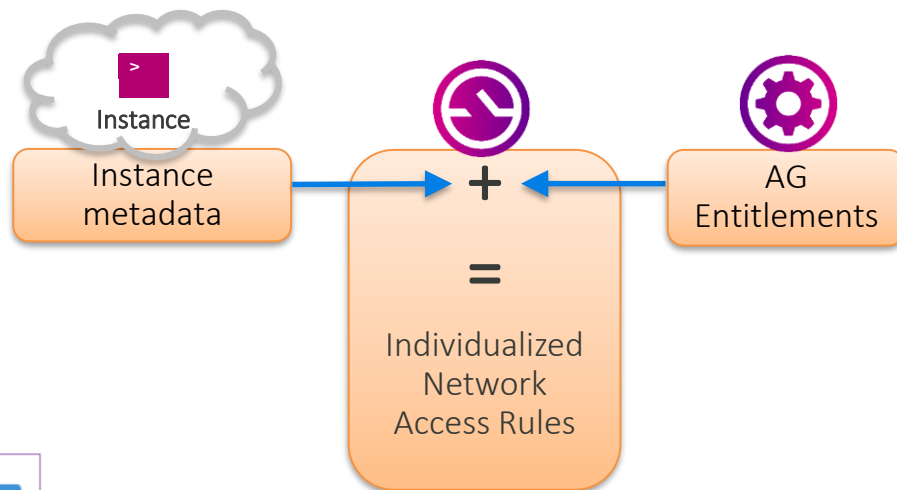
Moving from 4G to Wi-Fi states are sent to any Gateway (even the original) when connectivity is re-established (by OS)



Dynamic resource resolving

Managing static firewall rules in today's fast moving environments is next to impossible!

- Multi site deployments
- Failover sites
- IP addresses changes
- Dev-Ops
- Auto-scaling
- Life cycle management



Name Resolvers		Add new
DNS resolver	Name London	
AWS Resolver	Name AWS Resolver 1	
Azure Resolver	Name Azure Resolver - UK South	

Actions		Add new
#0 - ALLOW TCP	up to azure://tag-value:external on ports 80, 22	
#1 - ALLOW ICMP	up to azure://tag-name:access on types 0-255	

Role based API driven administration

Delegated – multi role admin

Multi tenant / Multi Site

API based for dynamic configuration

Type and Target define the exact privilege of an Admin Role. Tags limit the scope of the privilege.

The 'Privilege' dialog box is shown with a close button (X) in the top right corner. It contains three main sections: 'Privilege Type' with a dropdown menu set to 'All'; 'Target Item' with a dropdown menu set to 'All'; and 'Scope of Privilege' with a checkbox labeled 'Specific Tags or Individual Items' which is currently unchecked. At the bottom of the dialog are three buttons: 'Delete', 'Cancel', and 'Update'.

The 'Editing Administrative Role' page is displayed under the 'Admin Roles' tab. It features a navigation bar at the top with links to 'Appliances', 'Sites', 'IP Pools', 'Identity Providers', 'MFA Providers', and 'Admin Roles'. The main content area includes a back arrow and the title 'Editing Administrative Role'. Below this, there are three sections: 'Name' with a text input field containing 'Restricted Admin Role'; 'Notes' with a text area containing 'Admin role with restricted admin privileges'; and 'Privileges' with a list of three items: 'Create on Entitlement', 'View on Entitlement', and 'View on Policy'. At the bottom, there is a 'Tags' section with a text input field containing 'restricted_admin'.

AppGate SDP Introduction

The end...

Cyxtera