Identity-based Cross-cluster Fabrics

Igor Tarasenko, Co-founder & CTO, Bayware
Computation vs Networking

Common platform | Infra as code | Agility | Service portability | Cross-domain
---|---|---|---|---
Linux | Virtualization | DevOps/CICD | Containers | Any cloud
1990s | 2000s | 2005-10 | 2010s | 2018→

SDN | VNFs/Vendor-specific APIs | 2010s | 2015→

COMPUTATION → NETWORKING

Service Mesh

Identity-based cross-cluster fabrics | Igor Tarasenko, Co-founder & CTO | April 2019 | www.bayware.io
DevOps Desire the Declarative Model in Network

Provide applications instant and transparent cross-domain networking while eliminating low-level and repetitive configuration of legacy objects

- DNS records
- IP addresses
- Endpoint ACLs
- Network segments
- Perimeter ACLs
- Routes
- Tunnels
- Log & telemetry collectors
6 Great Leaps by Service Mesh for DevOps

Application-level networking on L4-7

- Software only overlay… infrastructure independent
- Every application gets its own network… based on deployment manifest
- Identity-based address and security model… comprehensively secure
- Every workload gets an agent… nearly instant response to application
- Orchestrated model… simpler to implement than scripting CNF/VNFs
- Communications visibility from application’s view… useful to DevOps
Pile-up on the Road to Multi-cloud/cluster

So what becomes of L2-3?

• VLANs, VRFs, Subnets
• VXLANs
• CNIs for IPAM, ACLs, bridges
• NAT

• Firewalls
• BGP, Segment routing
• Network service headers
• VPN gateways
All Networking in L4-7?

L2-3 network could be flat – no services beyond simple forwarding…

**If**

- L4-7 proxies find a way to avoid becoming a jumble of CNF/VNFs
- All settings can be easily derived from the application manifest
- It can implement corporate intent with respect to flow-level security

**Then**

- All those L2-3 solutions can go away in a flat world
But... Who doesn’t love a flat world?

**CISO requirements**
- Every node, service, and endpoint is authenticated and authorized
- Only authorized and encrypted flows can exist in the network
- Corporate isolation policy compliance

**Leading Application requirements**
- Some applications can’t traverse another application, i.e. proxies
- Other applications don’t want to re-code to pass proxies
- And still other applications are optimized without proxy next to each microservice
Instead... Can L2-3 Networks Make a Leap?

What if L2-3 had attributes of service mesh?

• Complete network and security setup derived from deployment manifest, e.g. application service graph
• Workload itself can change network forwarding behavior, no ‘behind-the-scenes’ configuration
• All networking based on workload identity with RBAC and declarative policies, not IP addresses
• Flows set up automatically in a Linux-based overlay – policy distributed actively and in-band
• Interconnection fabric comprised of policy engines paired with virtual switches
• Network provides ubiquitous telemetry that is meaningful for applications
From Service Graph to Data Flows in Three Steps

- Describe infrastructure-agnostic network policy in the form of declarative service graph
- Deploy fabric of lightweight interconnected Linux-based policy execution nodes
- Distribute flow-specific policy to nodes to instantiate flow according to the service graph
Service Interconnection Fabric

Application Service Graph

Service Interconnection Fabric

Flow Instantiation

- Service authz & name resolution
- Address translation
- Endpoint protocol filtering
- Flow-level microsegmentation
- Perimeter security
- Policy-based routing
- Link encryption & VPN tunneling
- Telemetry & logs

Complete network and security setup derived directly from existing deployment manifest, e.g. application service service graph
Rewards

• DevOps empowered
  • **Faster deployment:** Shorten time for hybrid cloud networking and security
  • **CI/CD-level agility:** DevOps replicates networking into any staging and production in minutes
  • **Greater productivity:** End-to-end orchestrated and re-usable code
  • **More meaningful telemetry:** Using application point of view

• Fully infrastructure agnostic – Deploys to any private or public cloud
• Pervasive security – Eliminate errors via automation of comprehensive application security
• Ease of use – Requires only application deployment manifest
• Simple – Even as it scales out
NORTH AMERICA

OPEN NETWORKING //
Enabling Collaborative Development & Innovation
How Bayware Works*

Kubernetes API Server

Policy Orchestrator

Kubernetes Worker Node

Processor Node (VM)

Identity-based cross-cluster fabrics | Igor Tarasenko, Co-founder & CTO | April 2019 | www.bayware.io

*Patent and patent pending