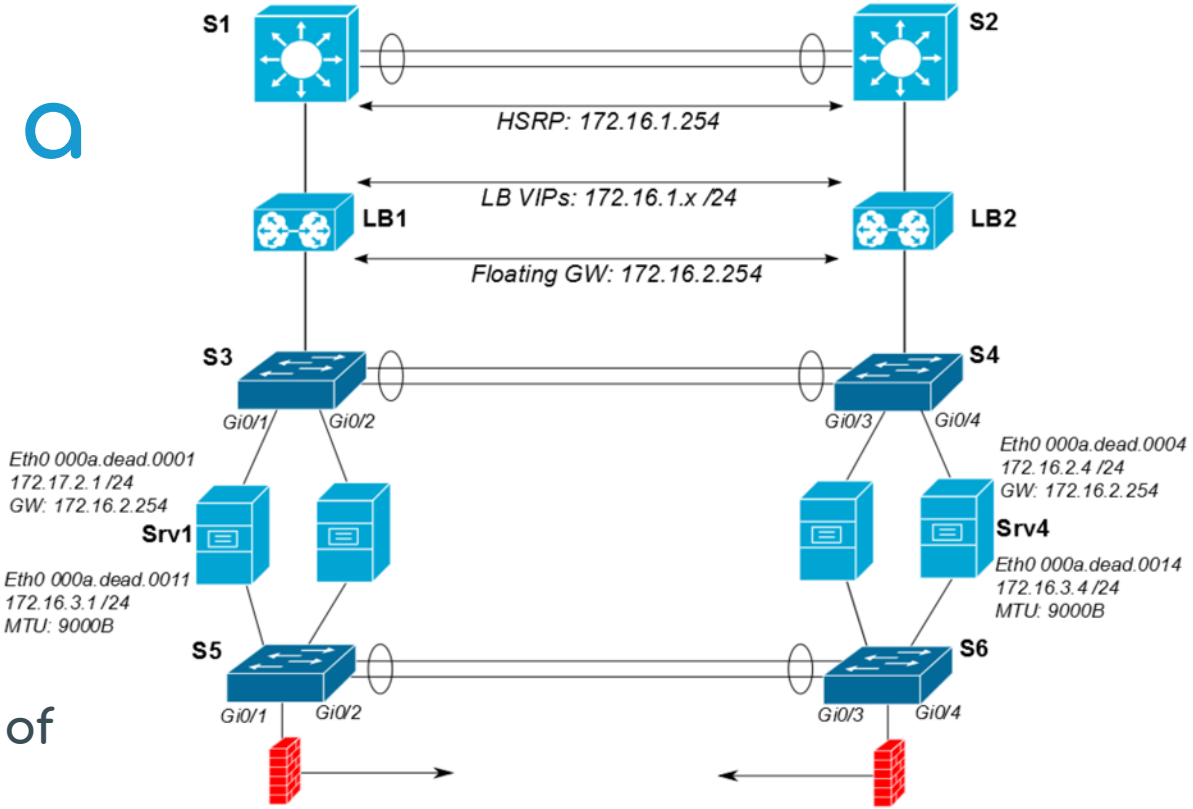
Docker Networking in Production at VISA

Sasi Kannappan
Chief Systems Architect, Visa Inc.

Mark Church
Solutions Architect, Docker Inc.

Can I get a VLAN?

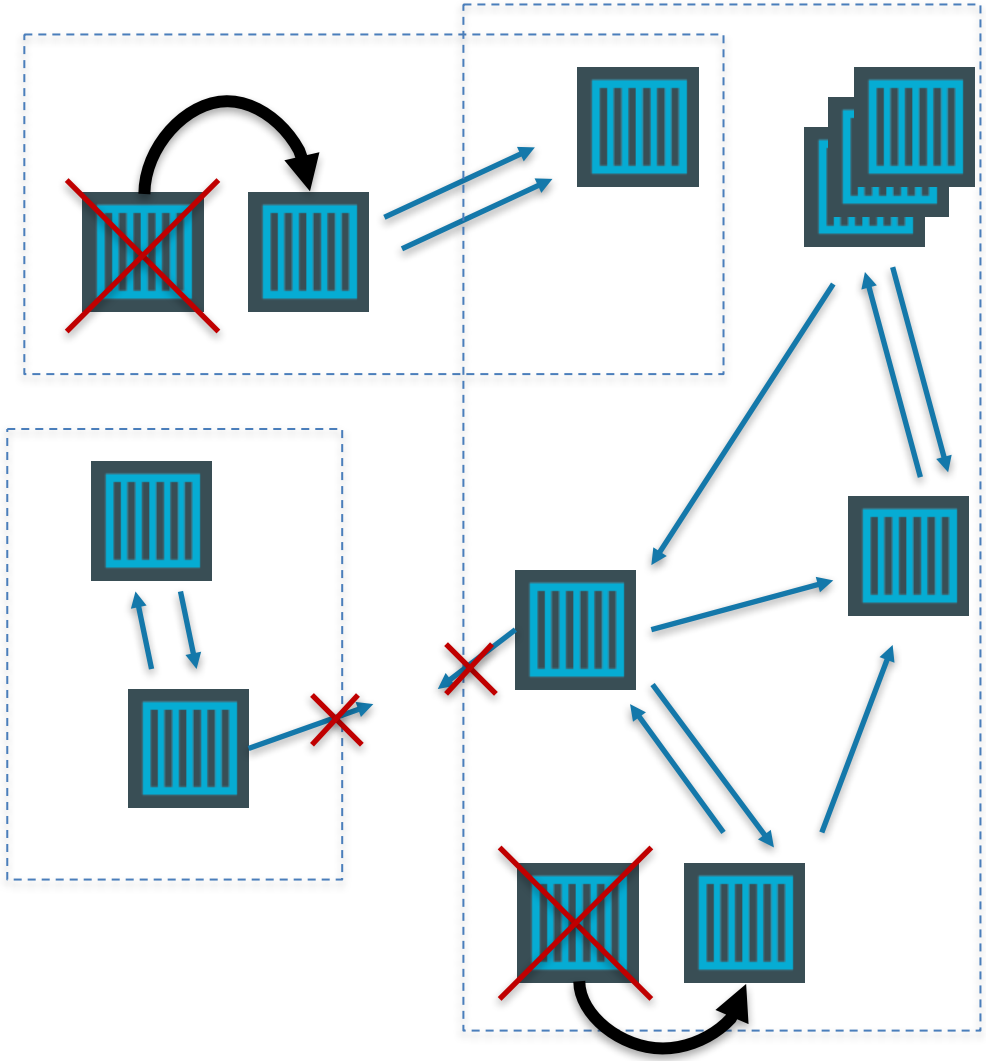
An abridged history of networking ...



Enter containers …

- 100s or 1000s of containers per host
- Containers that exist for minutes or months
- Microservices distributed across many more hosts (>>> E-W traffic)

… but this is worse.



Docker Networking

The Container Network Model (CNM)

Put Users and Applications First

Network policies defined in terms of applications

Plugin API Design

Batteries included but removable

Docker Networking

The Container Network Model (CNM)

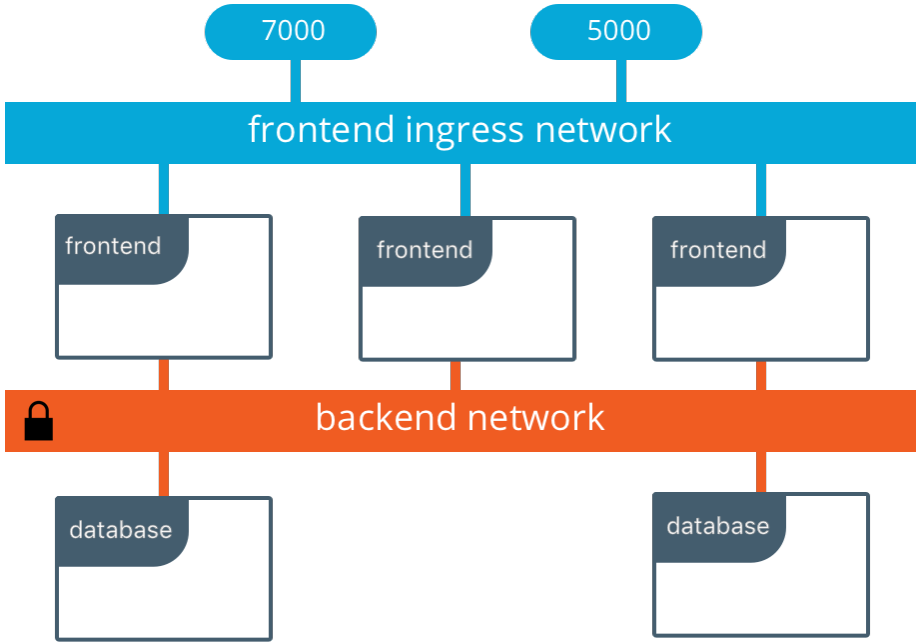
- Networks as a first-class citizen in Docker
- Scalable and secure control plane
- Pluggable network stack
- Support across OS ecosystem
- Easy multi-host networking

What is now possible?

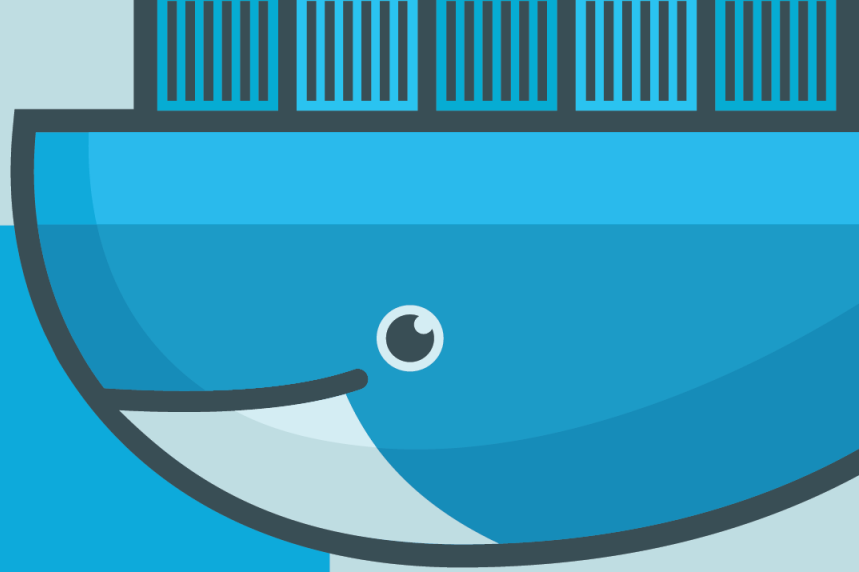
docker-stack.yml

```
services:
   frontend:
      networks:
         - ingress
         - backend
      ports:
        - 5000
        - 7000
   database:
      networks:
         - backend


networks:
   ingress:
      driver: overlay
   backend:
      driver: overlay
      driver_opts:
        encrypted : true
```



Docker Networking in Production at VISA: An Evolutionary Story



Sasi Kannappan

VISA

History of Docker at Visa

History

Started looking at Docker in late 2015
First containerized application in production in late 2016

Primary Application Environment

- A customer-facing financial transaction platform
- Now in production for 6 months
- Comprised of ~100 containers and ability to scale to ~800
- 2 production, 2 sandbox clusters, across 2 regions

VISA

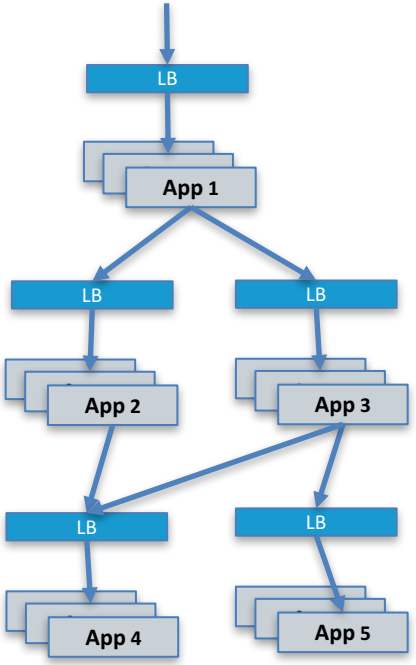
docker17 con

The Vision

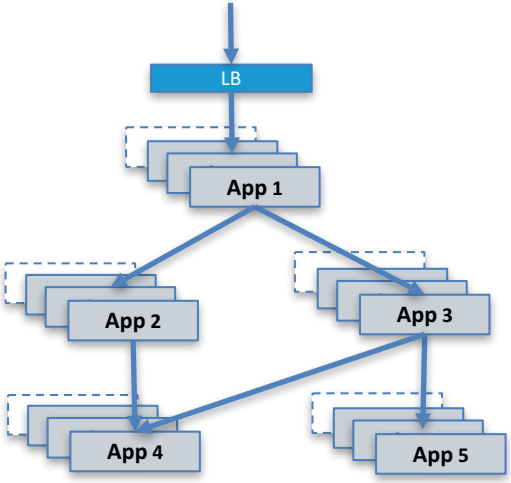
Goals

- Migration to Microservices Arch
- Dynamic scalability
- Operational simplicity
- Load balancer-less

From this …

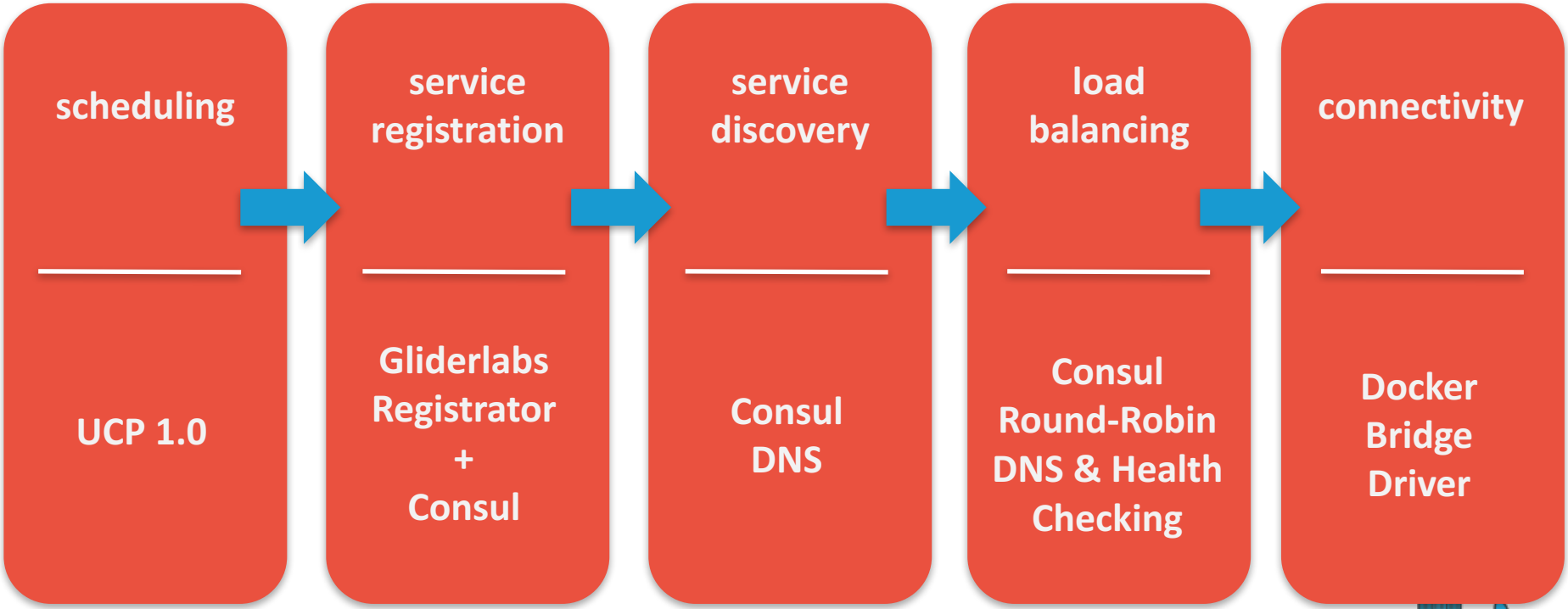


to this …



VISA

1st Gen Container Networking



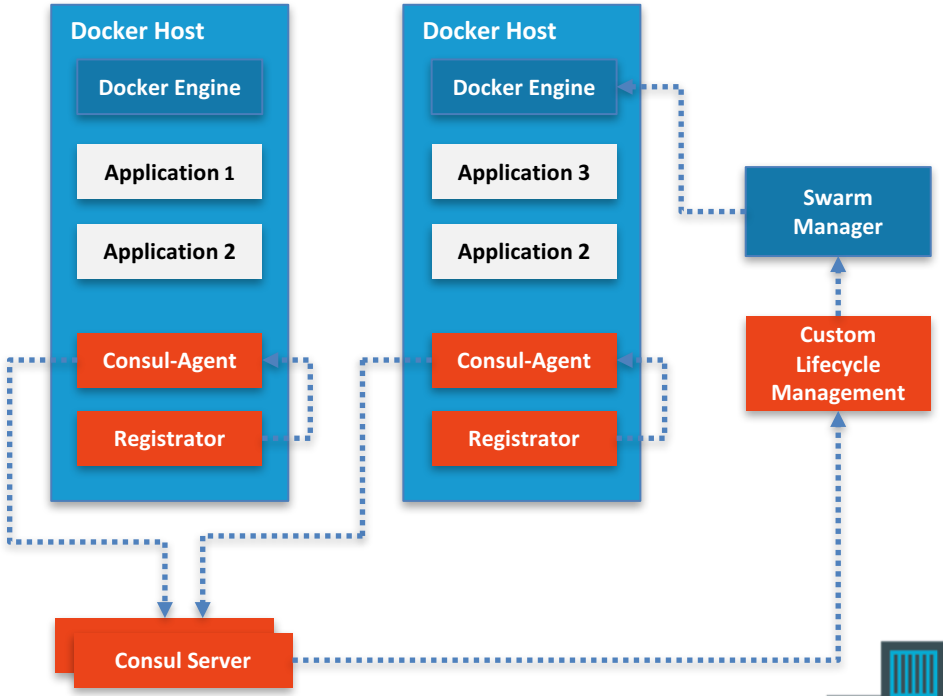
VISA

docker17 con

Service Registration

- Services and their location (ip:port) are registered centrally
- 1st gen: handled by registrator and consul

1st Gen Docker Architecture



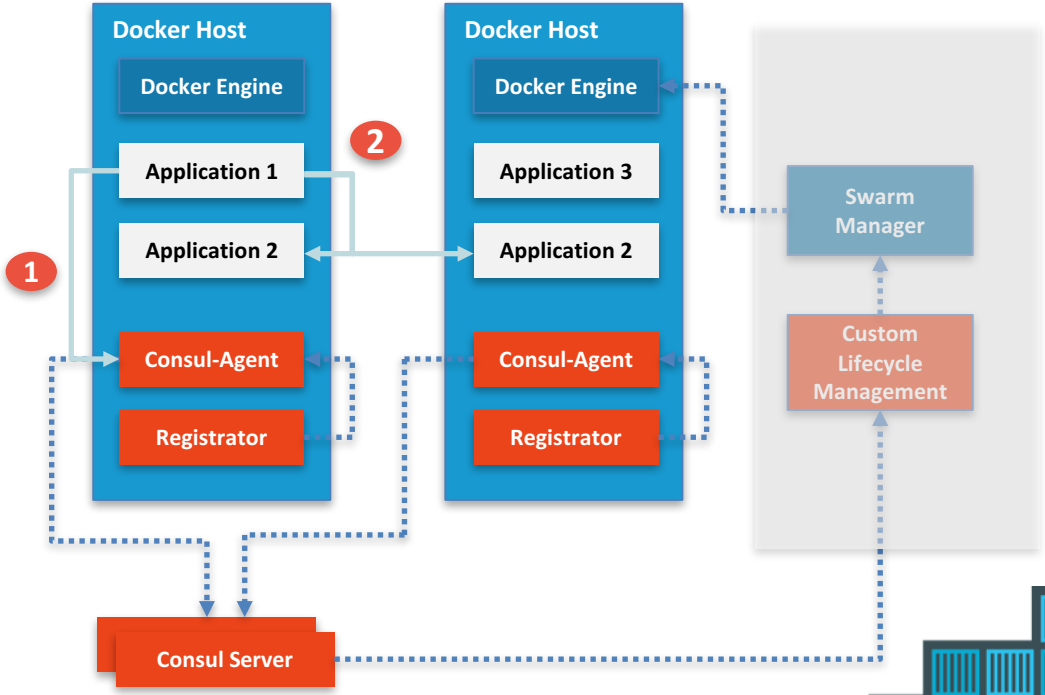
* Docker Bridge NW

VISA

Service Discovery

- The ability for services to find each other
- Designed using Consul capabilities

1st Gen Docker Architecture

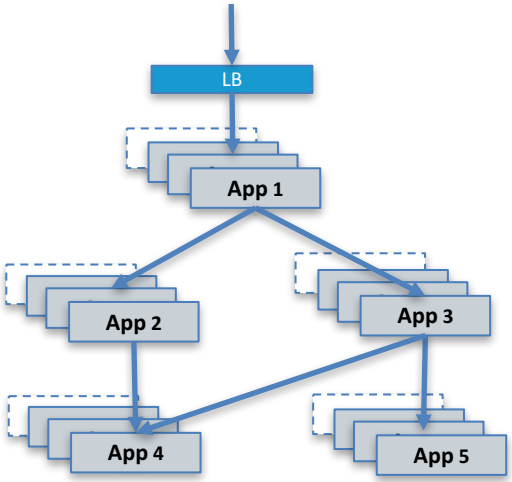


* Docker Bridge NW

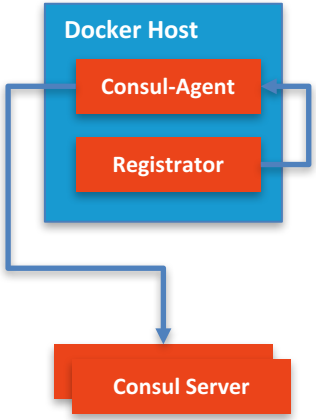
Problems with 1st Gen Architecture

- Complexity
  - Many components to manage
  - Maintaining HA for all components
  - Many component integrations to manage
  - Difficulty in troubleshooting
- Maintainability
  - Custom glue-code to manage

This is the goal …



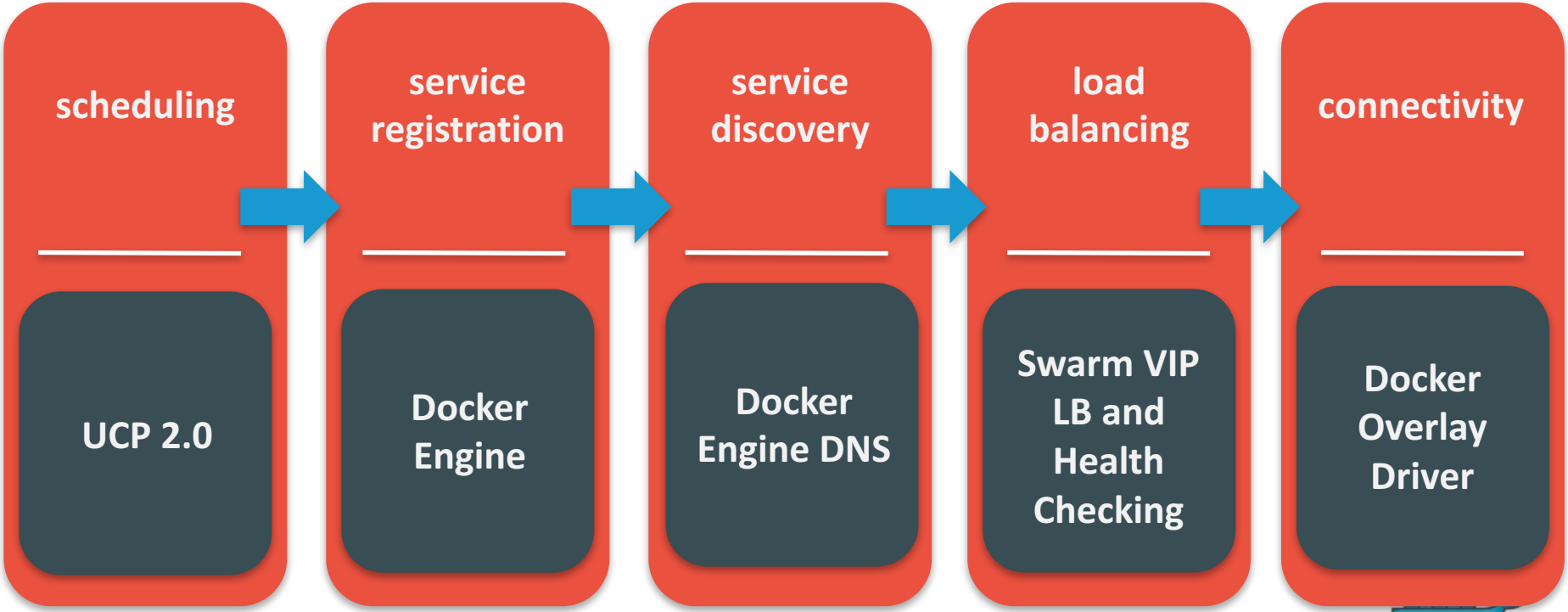
but, we replaced LB with this …



VISA

docker17 con

2nd Gen Container Networking



VISA

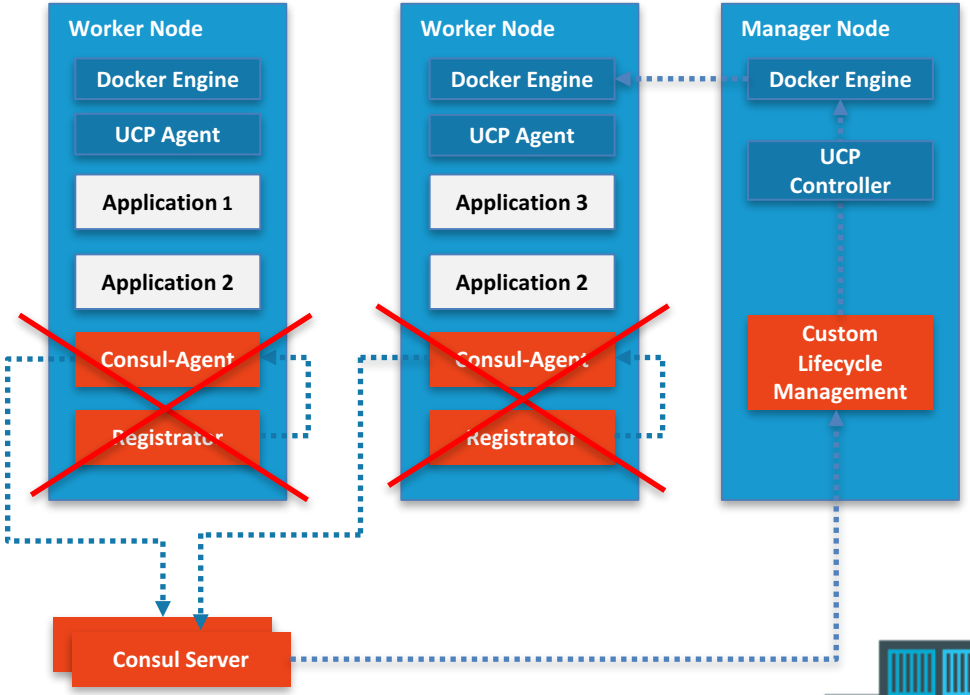
docker17 con

Service Registration & Service Discovery

- Service Definitions in Docker (Built-in registry)
- Docker Overlay NW
- VIP for Services

VISA

2nd Gen Docker Architecture

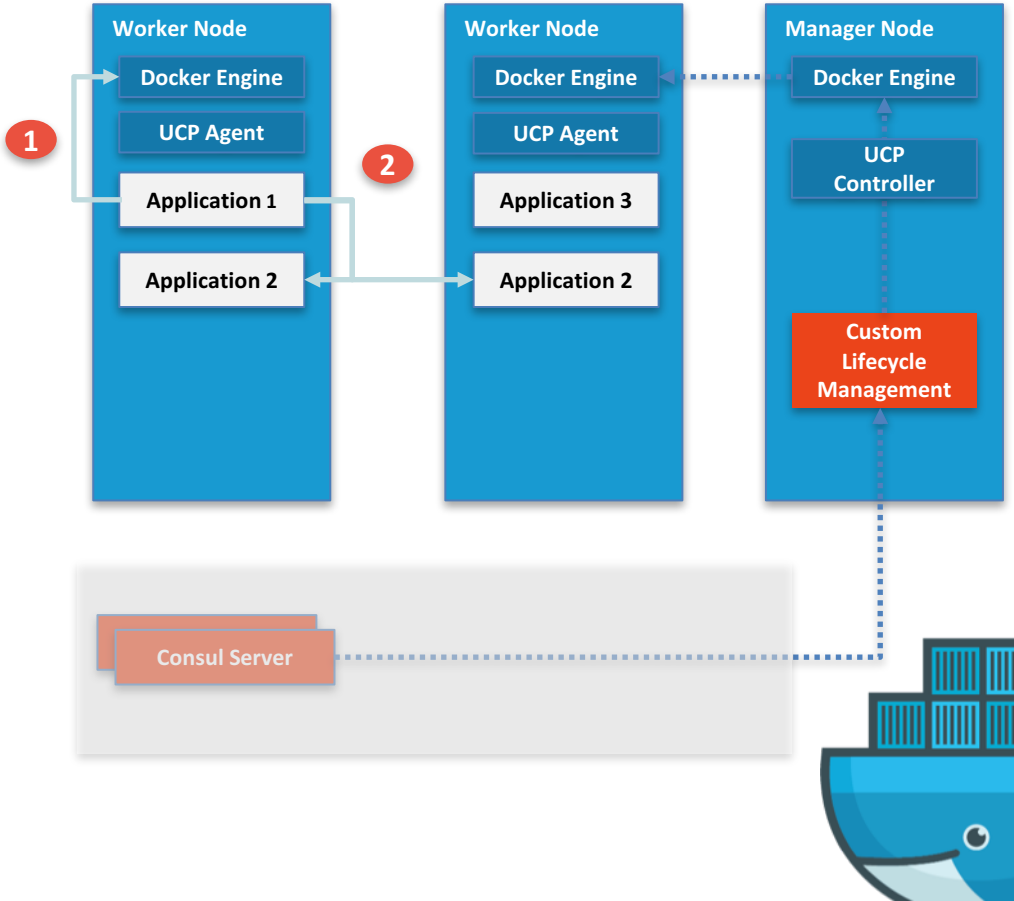


Load Balancing & Lifecycle Management

- Transparent to Applications
- Use VIPs
  - application1 (LB to Application2 containers on Nodes 1 & 2)
- http(s)://application1/

VISA

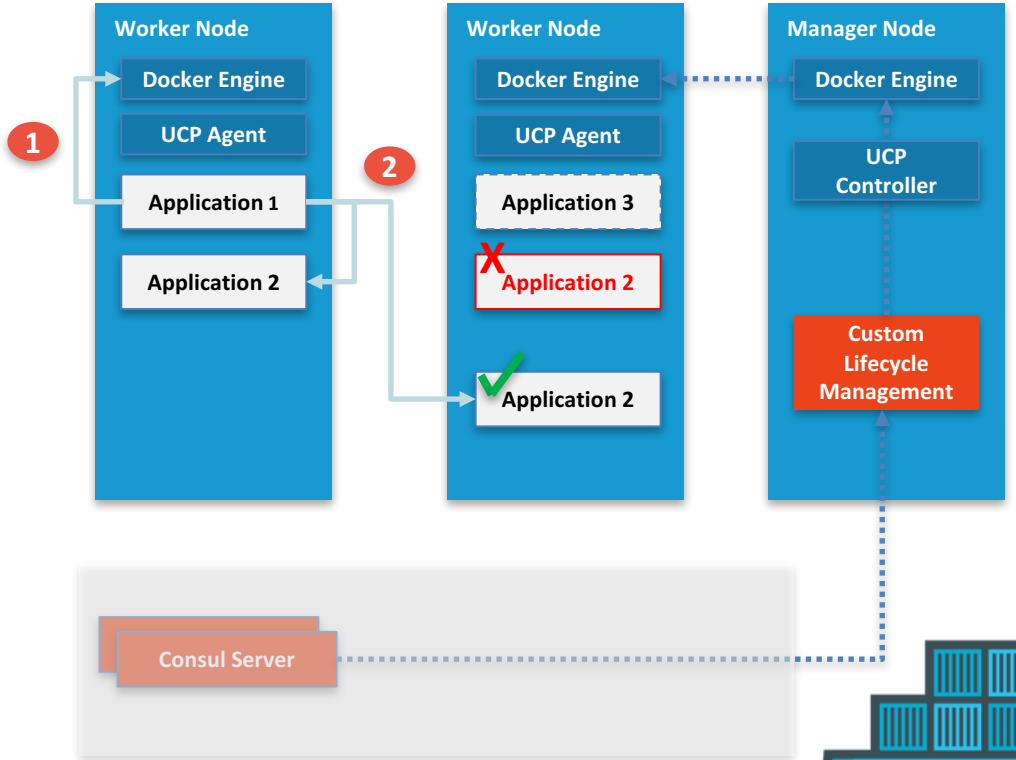
2nd Gen Docker Architecture



Load Balancing & Lifecycle Management

- Integrated Health check
- Self Heal Container Instances

2nd Gen Docker Architecture

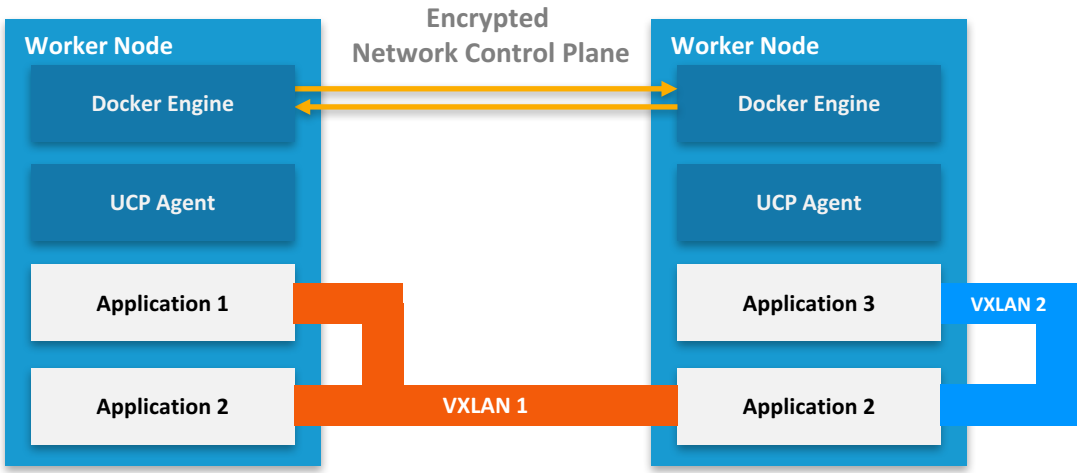


VISA

2nd Gen Docker Architecture

Connectivity

- VXLAN Overlay
- Transparent Overlay Encryption



VISA

Summary

- Less complexity in design -> easier to troubleshoot & better visibility
- Easier to maintain -> less cycles in upgrades and in
- Less custom code & fewer integration points
- Enhanced features (VIP load balancing and lifecycle mgmt)

DEMO TIME!

Thank you!

@docker #dockercon