



NGINX

August 2017

NGINX

Charles Pretzer

Technical Architect,
NGINX

MORE INFORMATION AT
[NGINX.COM](https://nginx.com)



About NGINX...

It all started with Igor





Building a Secure, Performant Network for Microservice Applications

August 8, 2017

NGINX

Agenda

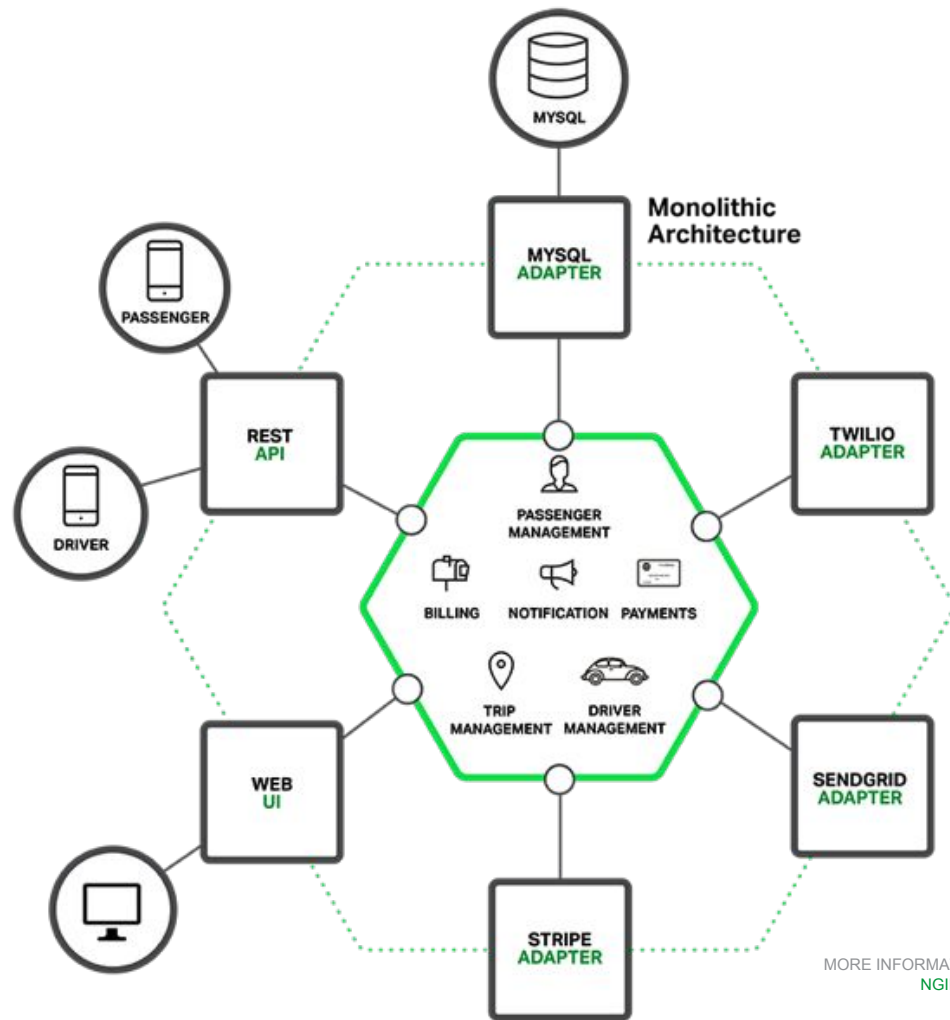
- The Big Shift
- The Networking Problem
 - Service Discovery
 - Load Balancing
 - Secure & Fast communication between microservices
- Architectures
- Q & A

The Big Shift

MORE INFORMATION AT
[NGINX.COM](https://nginx.com)

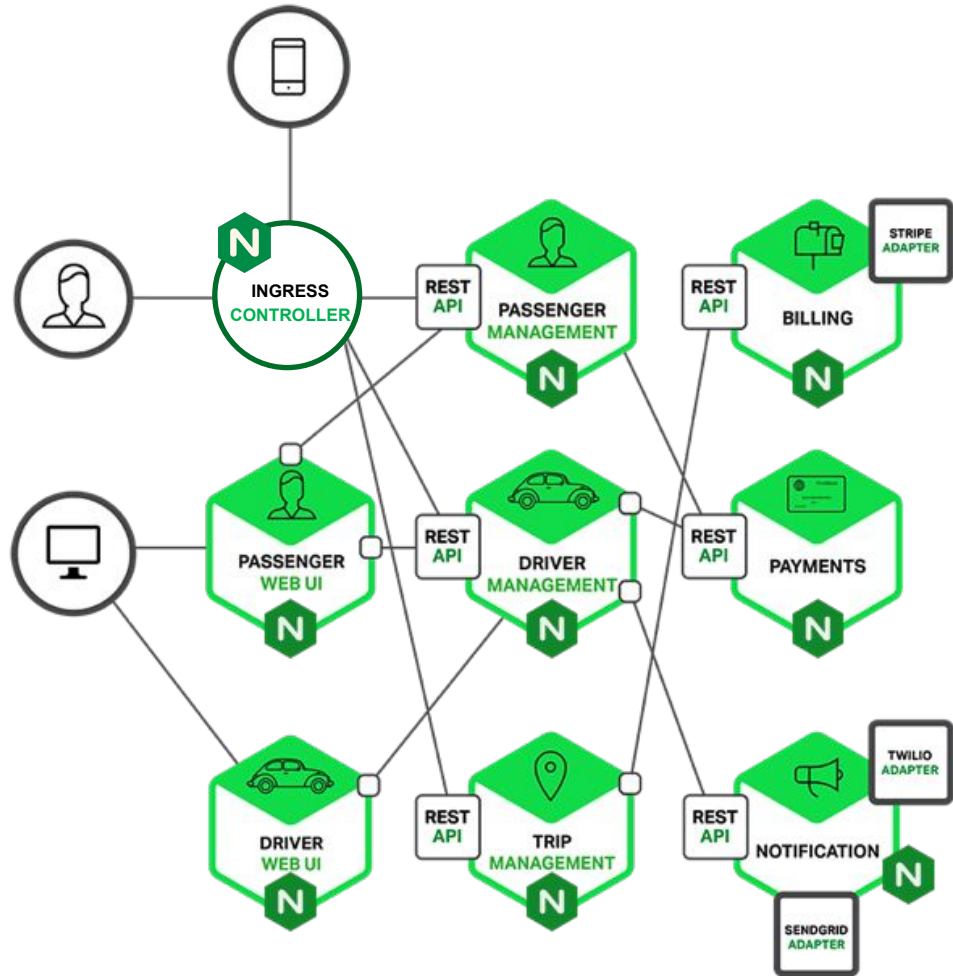
Architectural Changes:

Monolith



Architectural Changes:

Monolith ... to Microservices

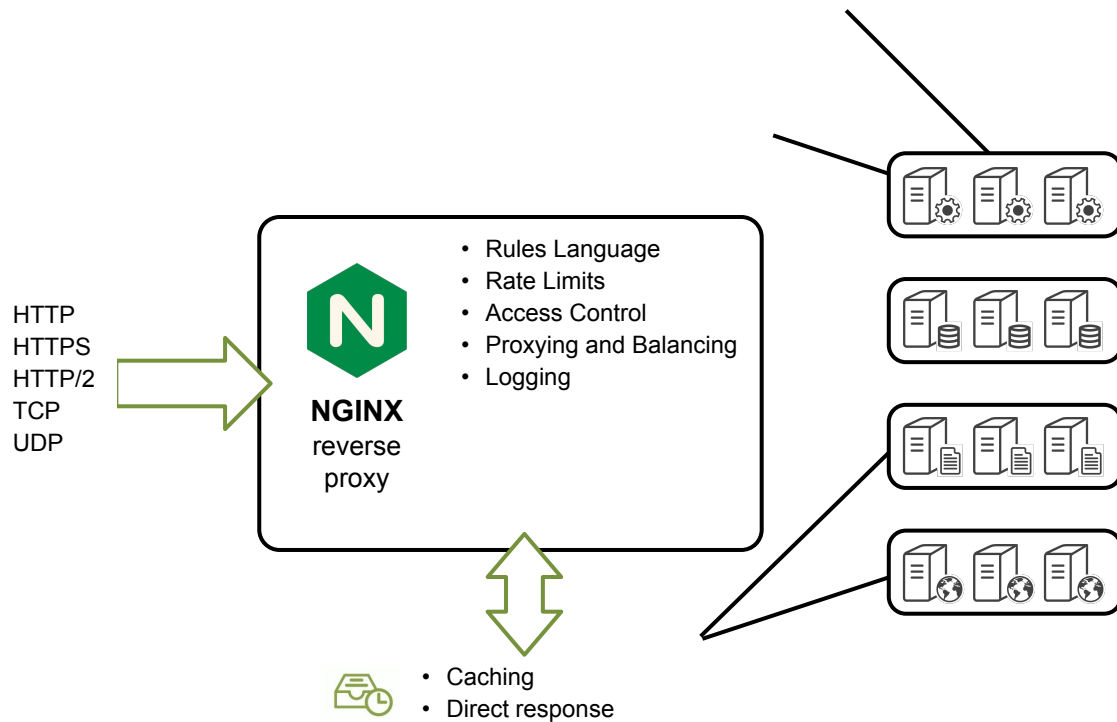


MORE INFORMATION AT
[NGINX.COM](https://www.nginx.com)

NGINX ❤️ Microservices

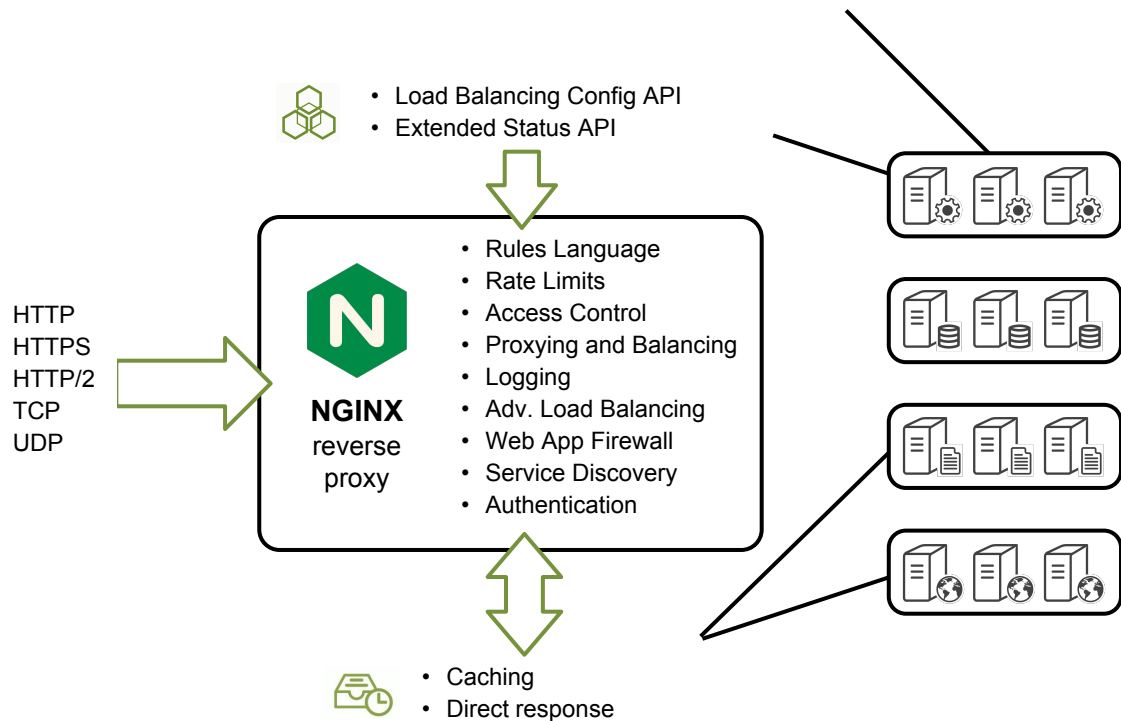
What is NGINX?

NGINX



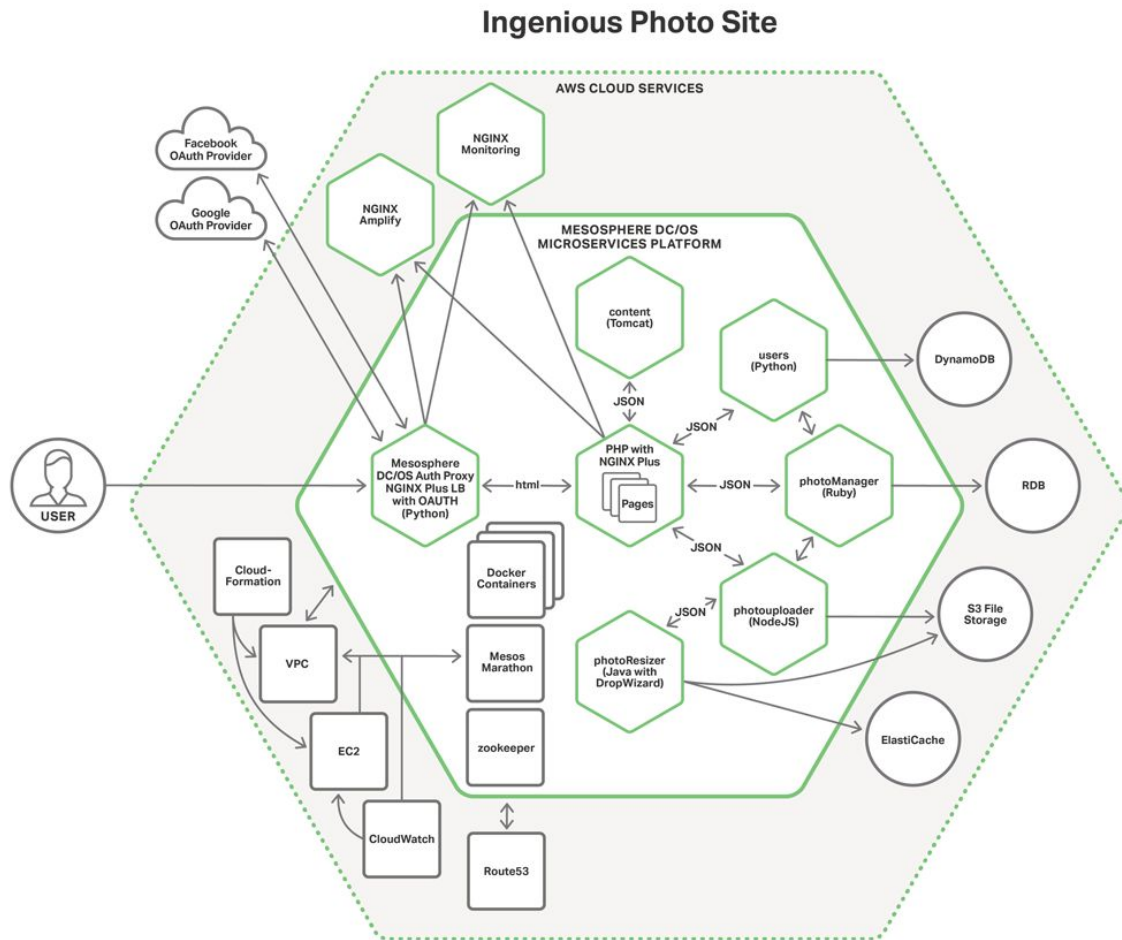
What is NGINX

...and NGINX Plus?



Microservices Reference Architecture

- Docker containers
- Polyglot services
- 12-Factor App(-esque) design

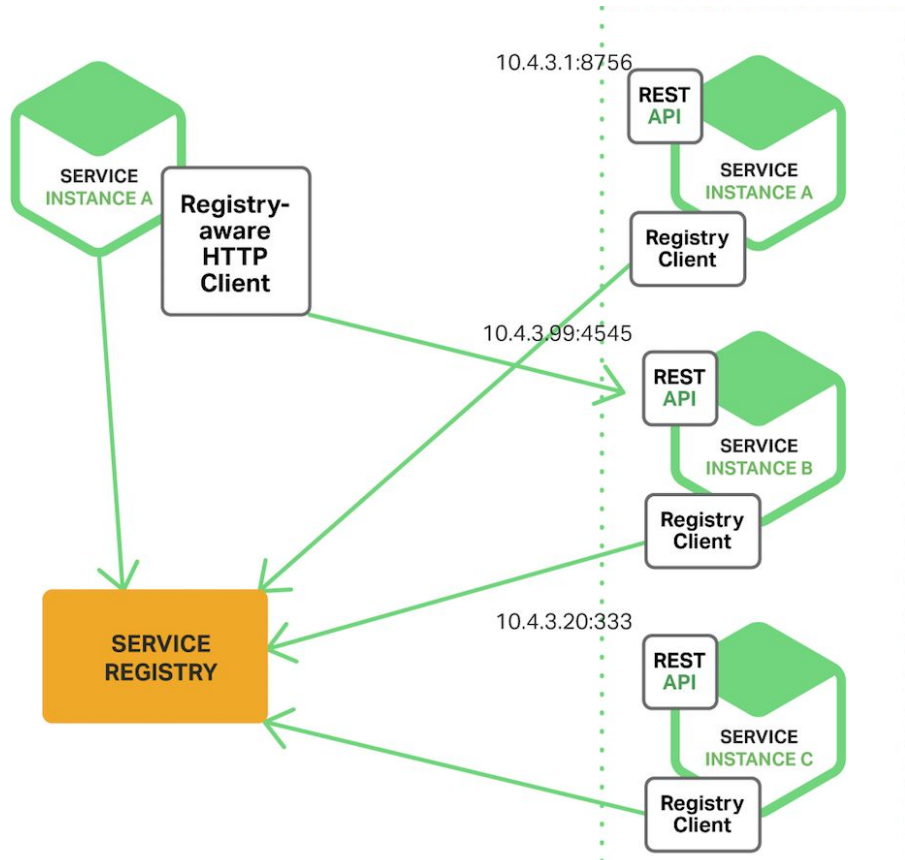


The Networking Problem

MORE INFORMATION AT
[NGINX.COM](https://nginx.com)

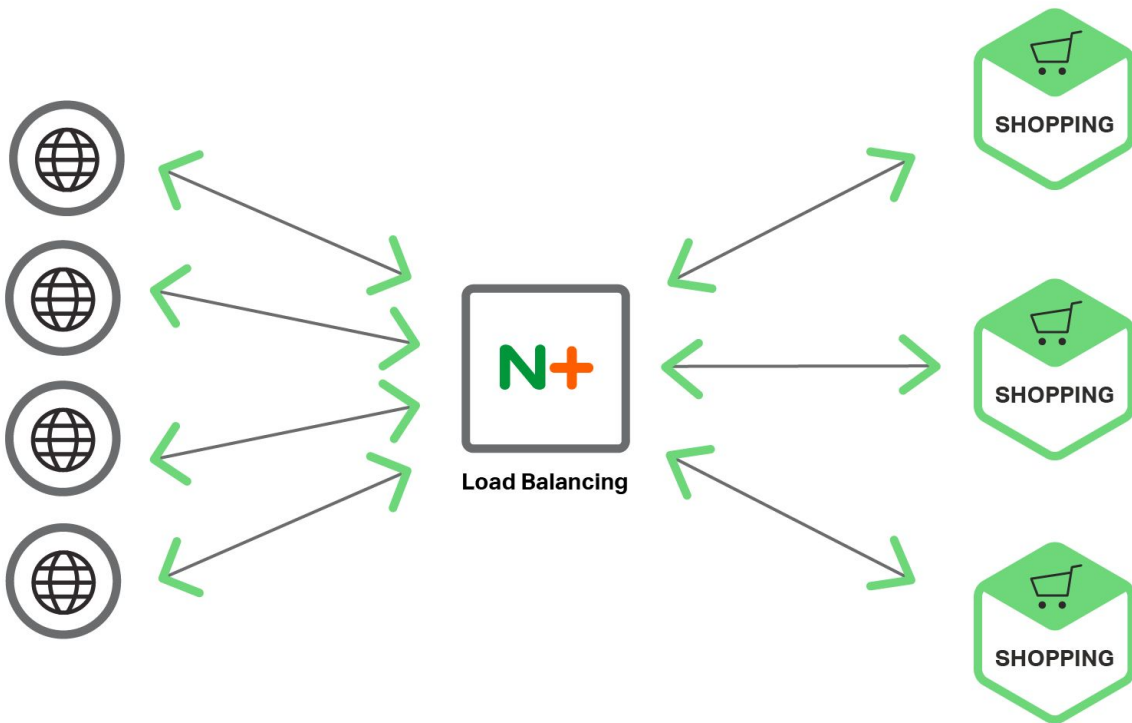
Service Discovery

- Services need to know where other services are
- Service registries work in many different ways
- Register and read service information



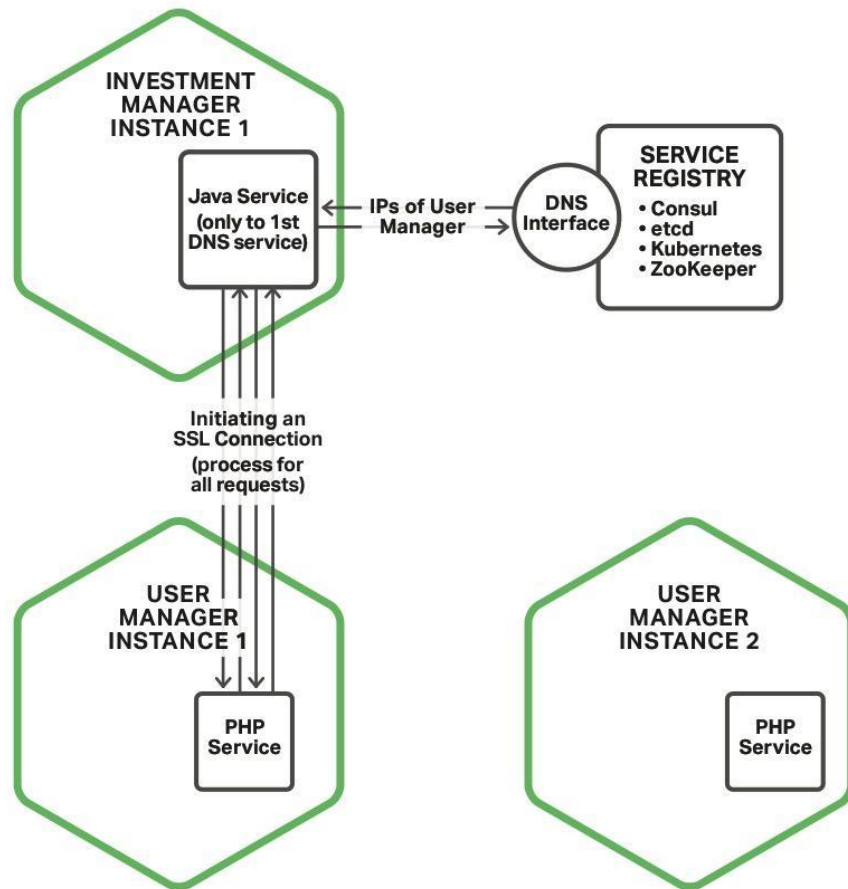
Load-balancing

- High Quality Load Balancing
- Developer Configurable



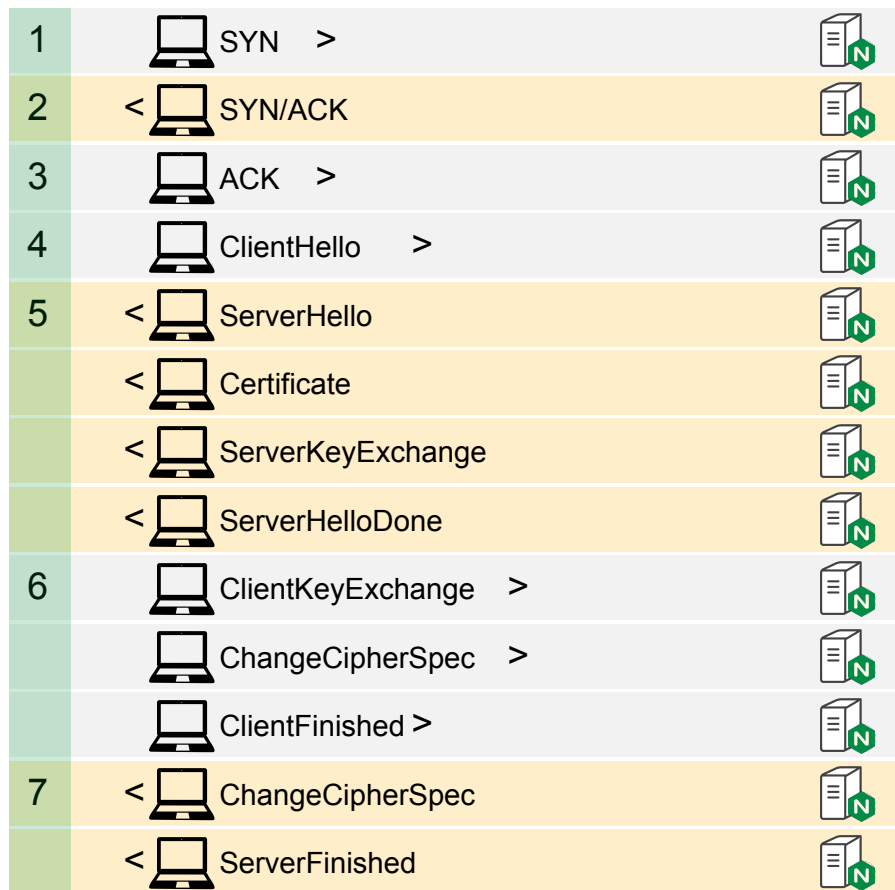
Secure & Fast Communication

- Encryption at the transmission layer is becoming standard
- SSL communication is slow
- Encryption is CPU intensive



The SSL problem

- A new SSL connection takes a minimum of 7 messages to establish



Solution

- **Service discovery**
- **Robust load balancing**
- **Fast encryption**

Network Architectures

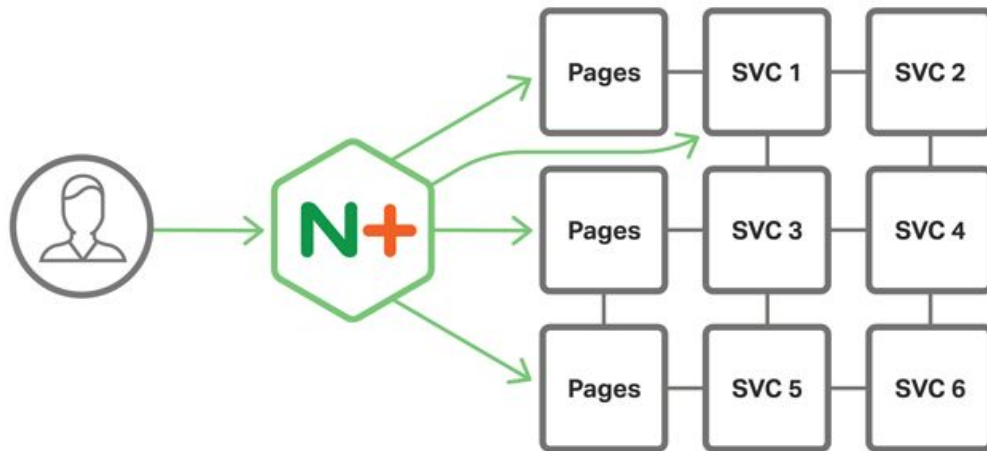
MORE INFORMATION AT
[NGINX.COM](https://nginx.com)

Proxy Model

- Focus on internet traffic
- A shock absorber for your app
- Dynamic connectivity

Proxy Model

- Inbound traffic is managed through a reverse proxy/load balancer
- Services are left to themselves to connect to each other.
- Often through round-robin DNS

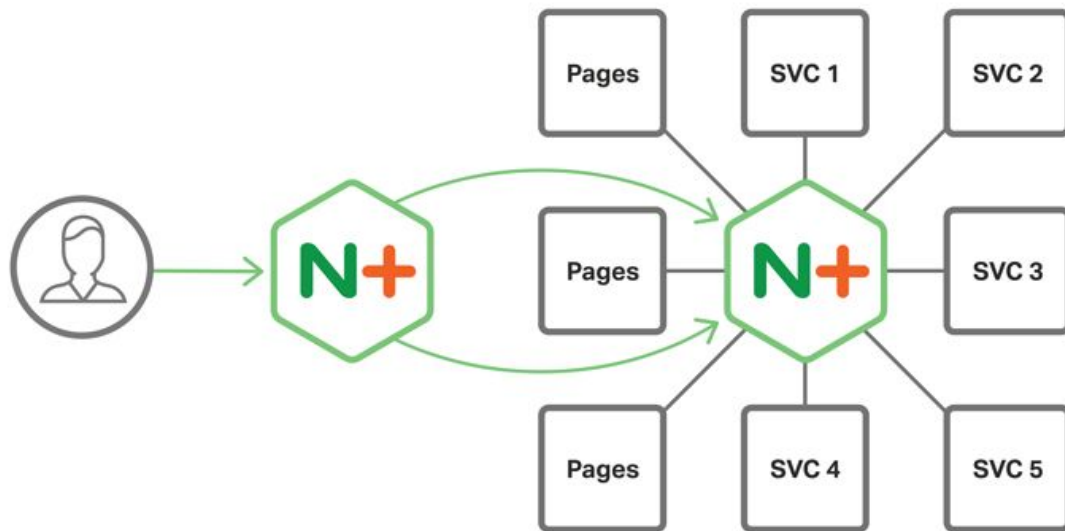


Router Mesh

- **Robust service discovery**
- **Advanced load balancing**
- **Circuit breaker pattern**

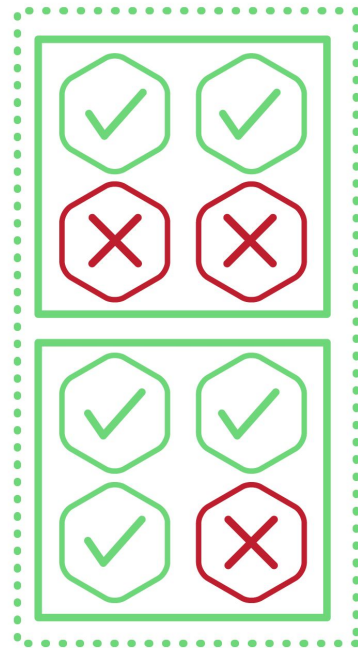
Router Mesh Model

- Inbound routing through reverse proxy
- Centralized load balancing through a separate load balancing service
- Deis Router works like this



Circuit Breakers

- Active health checks
- Retry
- Caching



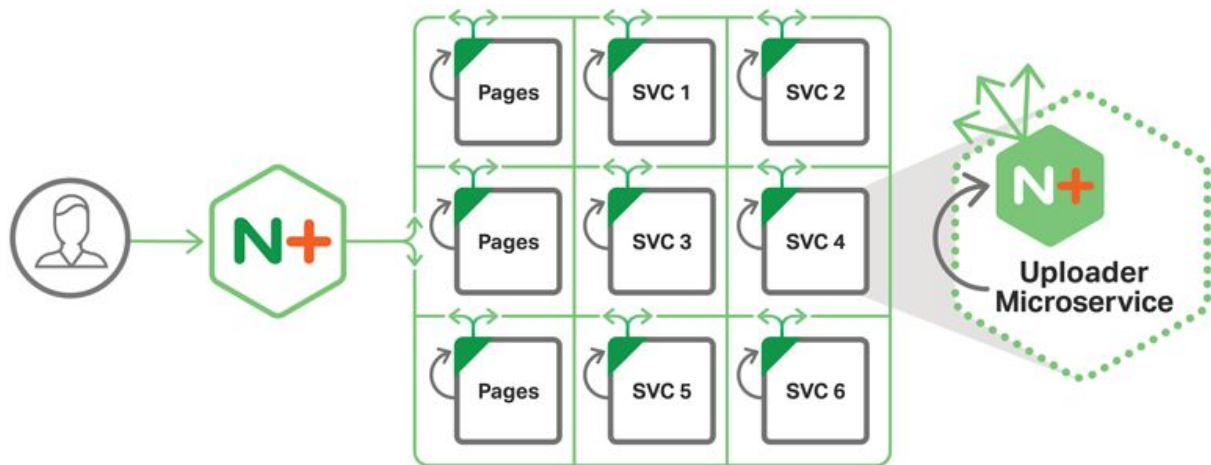
Fabric Model

- Robust service discovery
- Advanced load balancing
- Circuit breaker pattern
- Persistent SSL network

Inter-Process Communication

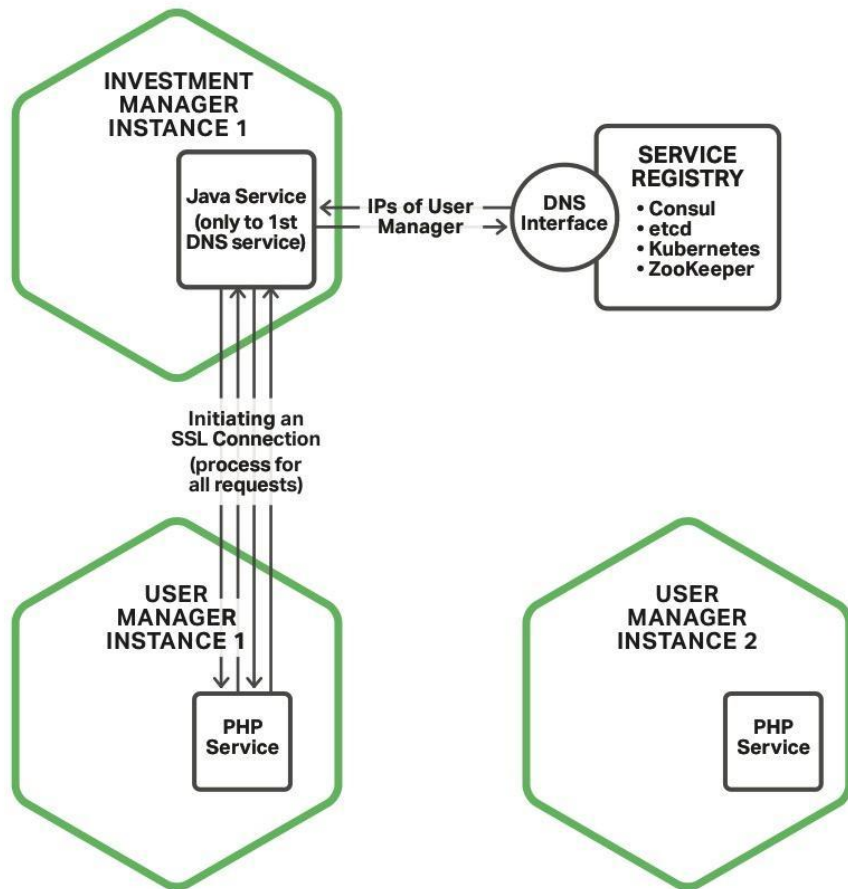
- Routing is done at the container level
- Services connect to each other as needed
- NGINX Plus acts as the forward and reverse proxy for all requests

Fabric Model (e.g. Mesos)



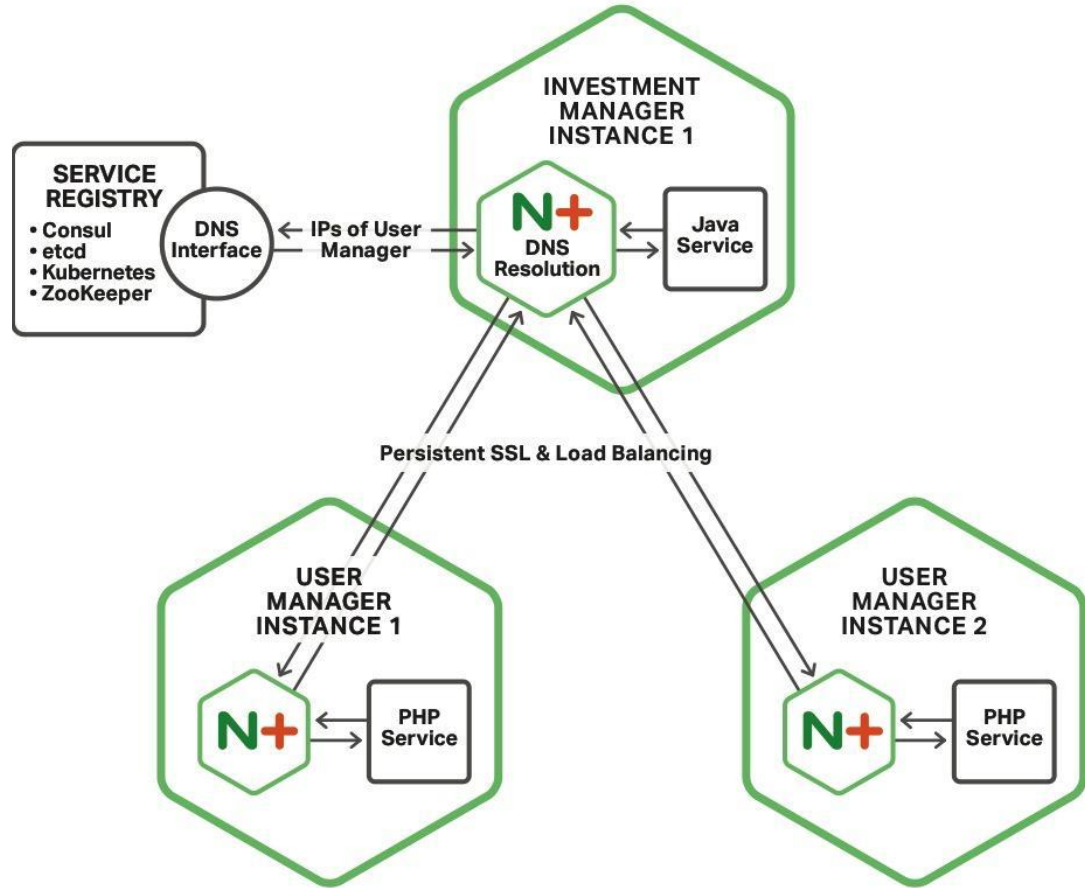
Normal Process

- DNS service discovery
- Relies on round robin DNS
- Each request creates a new SSL connection which fully implemented in 7+ requests



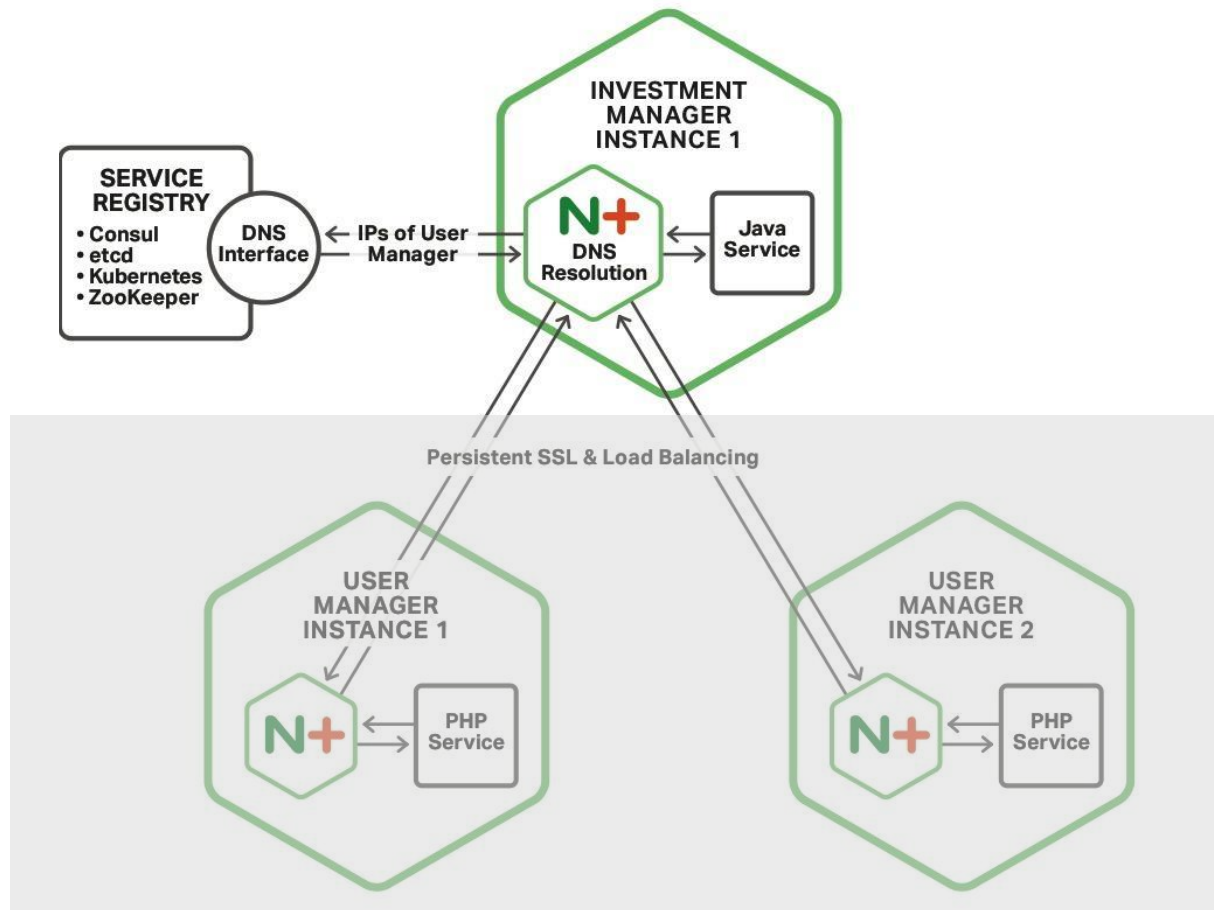
Detail

- NGINX Plus runs in each container
- Application code talks to NGINX locally
- NGINX talks to NGINX
- NGINX queries the service registry



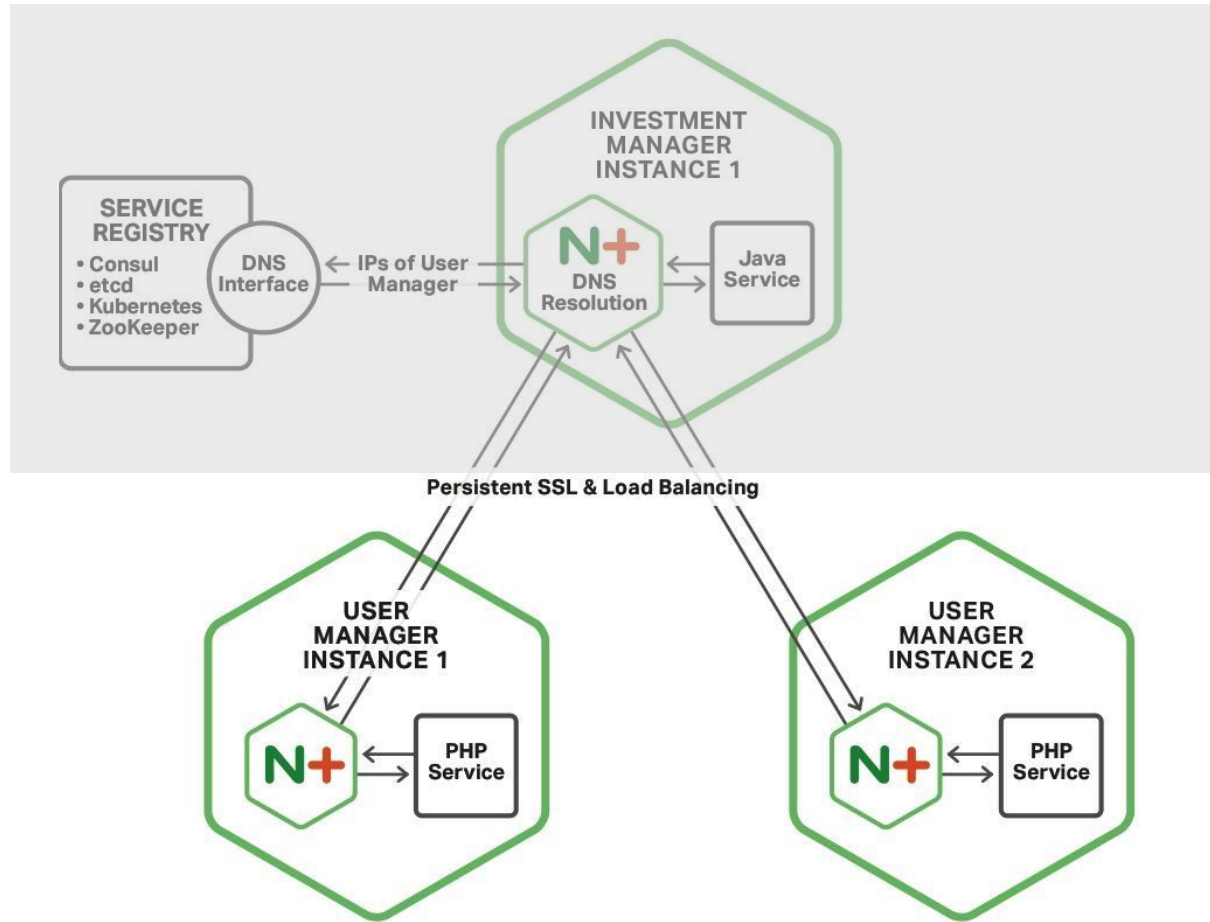
Service Discovery

- DNS is a clear way to manage service discovery
- NGINX Plus Asynchronous Resolver
- SRV records allow you to effectively use your resources



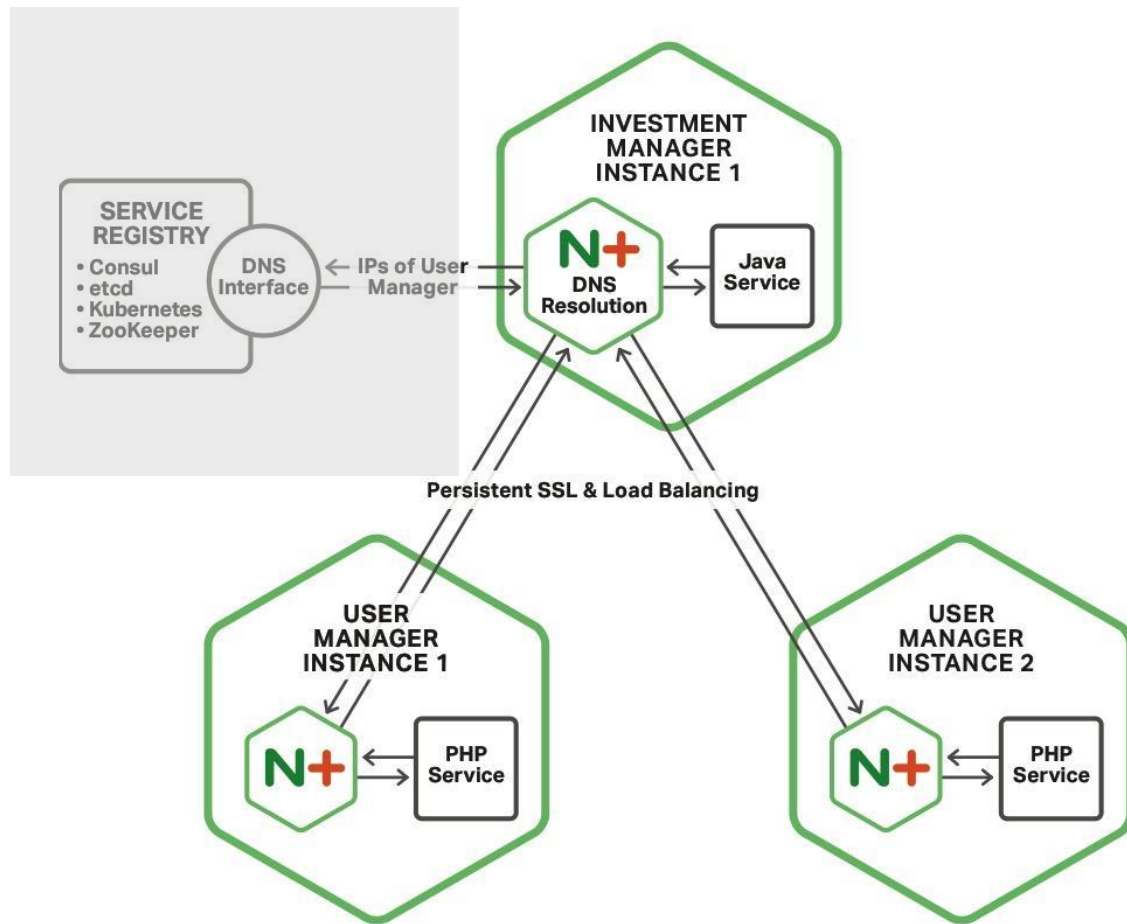
Load-balancing

- Proper request distribution
- Flexibility based on the backing service
- Different load-balancing schemes



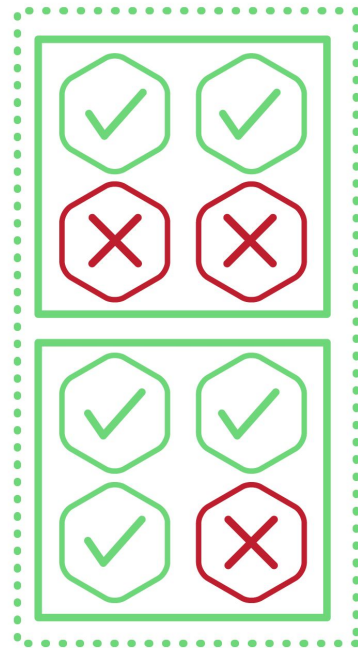
Persistent SSL Connections

- Applications generate thousands of connections
- 7+ steps in SSL negotiation
- Persistent SSL upstream keepalive



Circuit Breaker Plus

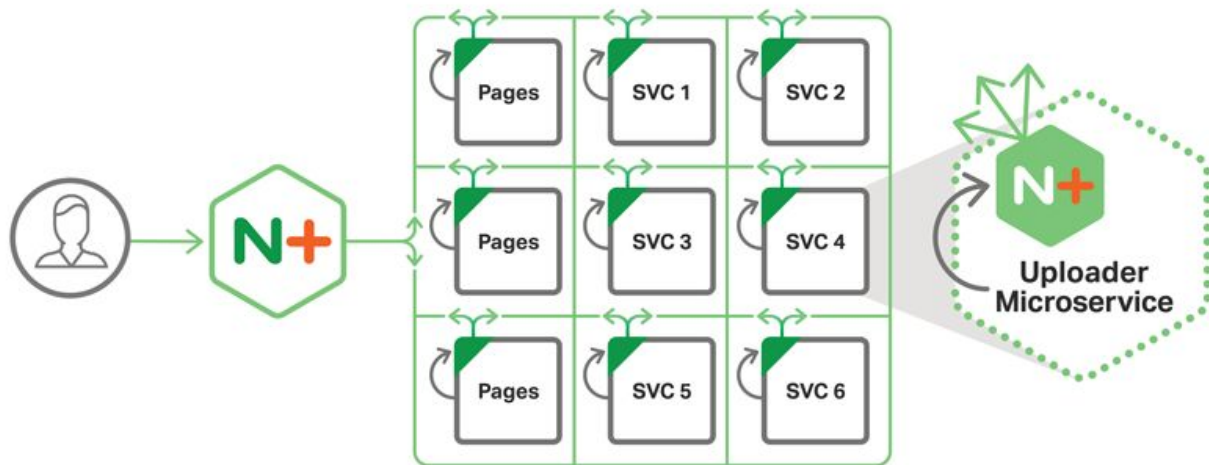
- Active health checks
- Retry
- Caching



The solution

- Service discovery
- Container-based load-balancing
- Persistent SSL connections
- Circuit-breaker functionality
- Status data on both sides of the equation

Fabric Model (e.g. Mesos)



Resources

MORE INFORMATION AT
[NGINX.COM](https://nginx.com)

Web Resources

Reading Materials

- www.nginx.com/blog/microservices-reference-architecture-nginx-fabric-model/
- www.nginx.com/resources/library/microservices-reference-architecture/

Fabric Model Architecture Repository

- <https://github.com/nginxinc/fabric-model-architecture>

Monitoring Demo

- <https://demo.nginx.com>

Q & A

MORE INFORMATION AT
[NGINX.COM](https://nginx.com)

Closing Thoughts

- The Microservices Architecture is a powerful evolution of system design
- With great power comes great responsibility, so it's important to be aware of what's involved in a migration to the Microservices Architecture
- NGINX has designed three networking architectures for microservices which will help in migrating from a monolith or starting with a green field system design