The art of automating the automation

PayPal’s journey of Infrastructure automation

Agenda

- About PayPal
- PayPal Infrastructure Overview
- Domains & Automation needs
- Challenges & Opportunities
- Framework Development
- Results
- Closing thoughts

PayPal

Overview

Digital payments leader

PayPal has remained at the forefront of the digital payment revolution for more than 20 years by leveraging technology to make financial services and commerce more convenient, affordable, and secure

- Launched in 1999; Launched as independent public company in 2015
- ~430 million active accounts
- Available in 200+ markets
- 30,000+ employees

PayPal

A Global Infrastructure

Infrastructure Architecture

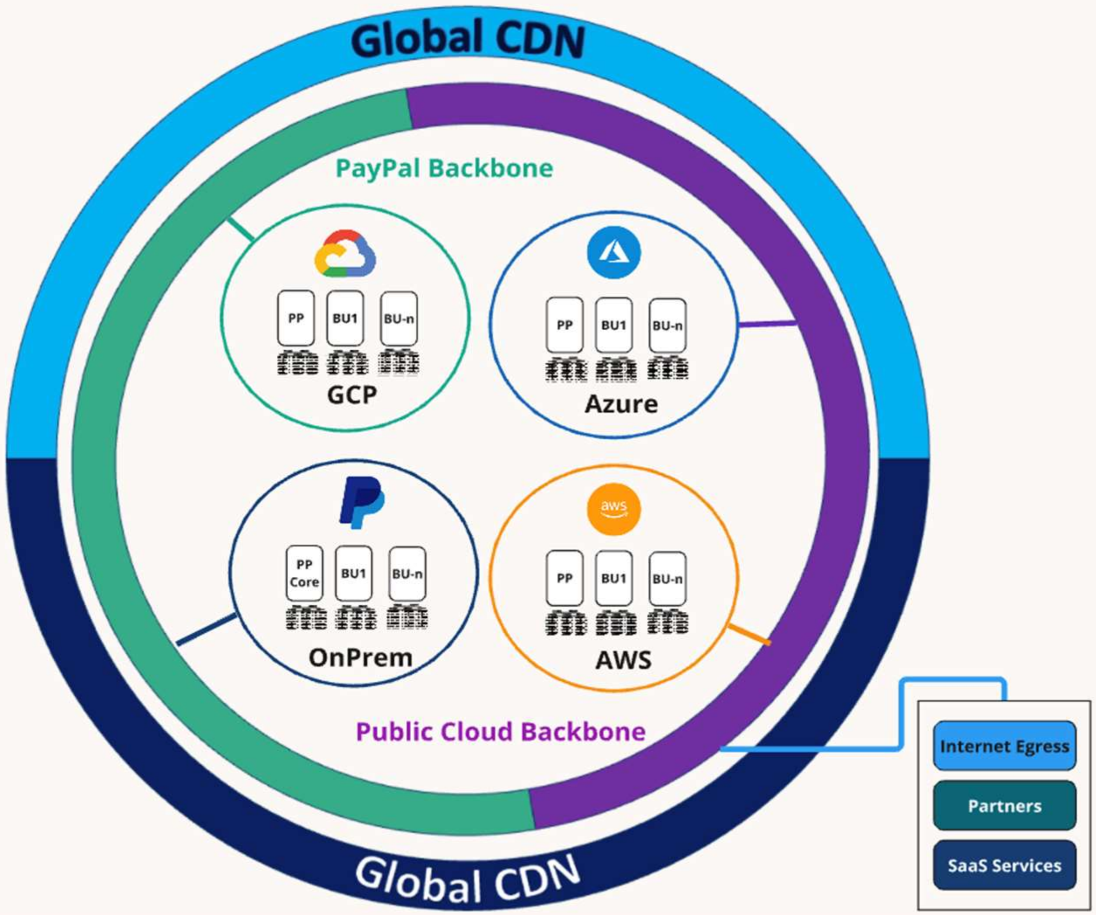
- Hybrid, Multi cloud, Multi BU, Multi Vendor Network

- Global CDN with commercial CDN providers
  - Multi vendor & diverse capabilities
  - Highly reliable at each POP level

- Mixed backbone strategy
  - PayPal as an ISP (Dark fiber, leased lines)
  - Public cloud backbone network

- Compute
  - All major cloud providers
  - Different BUs hosted on different cloud providers
  - Huge private datacenters for core
  - GPU racks for AI workloads

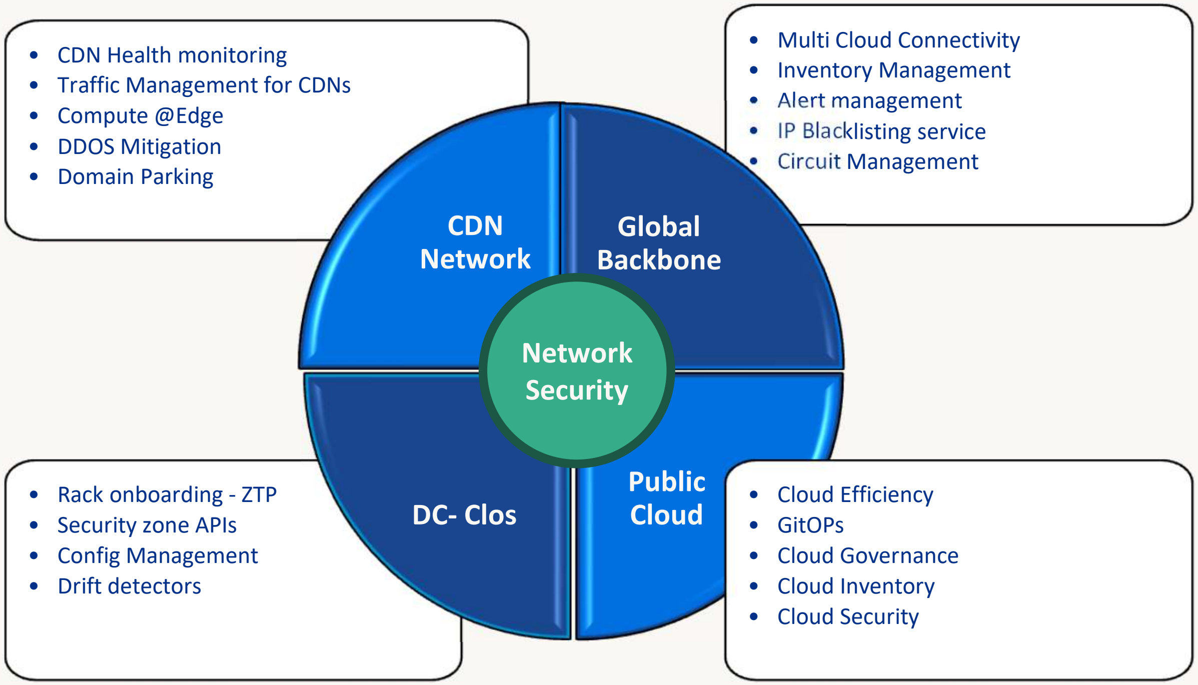
- Network & Security
  - Typical CLOS network with security zones & firewalls
  - Dual vendor strategy for network & security



PayPal

Multiple Complex Domains

- CDN Networks need autonomous & manual traffic management to mitigate attacks, availability issues
- We rely on CDN Network for reducing latency thru embracing Compute @ Edge
- Backbone network has security solutions that are automatic with security teams identifying rouge IPs
- Comprehensive alert and inventory management for backbone networks
- Multi Cloud connectivity solutions for backbone
- Complete git-ops mode for Public cloud.
- Inhouse solutions for cloud efficiency and accountability.
- GPU Scheduling services for AI Workloads
- Network automatic configuration for security zone movements
- Traditional firewall rules via self service & identity based firewall rules for workloads



CDN Network

- CDN Health monitoring
- Traffic Management for CDNs
- Compute @Edge
- DDOS Mitigation
- Domain Parking

Global Backbone

- Multi Cloud Connectivity
- Inventory Management
- Alert management
- IP Blacklisting service
- Circuit Management

Network Security

DC- Clos

- Rack onboarding - ZTP
- Security zone APIs
- Config Management
- Drift detectors

Public Cloud

- Cloud Efficiency
- GitOPs
- Cloud Governance
- Cloud Inventory
- Cloud Security

Challenges

- Ever increasing automation needs
- Tools need to be meeting reliability standards of the organization
- Security of the tools is as important as the organization’s product itself
- Velocity/Feature gap between public cloud & private cloud
- Automated changes go thru same wide range of internal processes
- Small teams & Limited budgets
- All needs were due yesterday
- Growing demand for self service tools

Opportunities

- Shift from network automation to platform engineering
- Enable rapid application development by embracing frameworks
- Platform engineering needs to follow scalable services architecture
- Reduce errors by automating the automation

Common & Repetitive needs

Need for a framework

- Each Infra/Network Service needs UI, Standard authentication, Database, Ticketing System etc
- Each enterprise have their own standards for authentication & record keeping etc
- Our experience says, It takes 150 working days for one service to be developed
- Such long lead cycle for a single servie is not acceptable
- We needed a framework to simplify the application development.
- Its an existing pattern in several areas
- Most enterprises have frameworks for their product applications.
- Android Application, iOS Application have succeeded with similar approach on a large
- Network automation needed to embrace similar thought process

UI Development

Authentication

DB Interface

Authorization

Service Accounts

Ticketing

Certs & VIPs

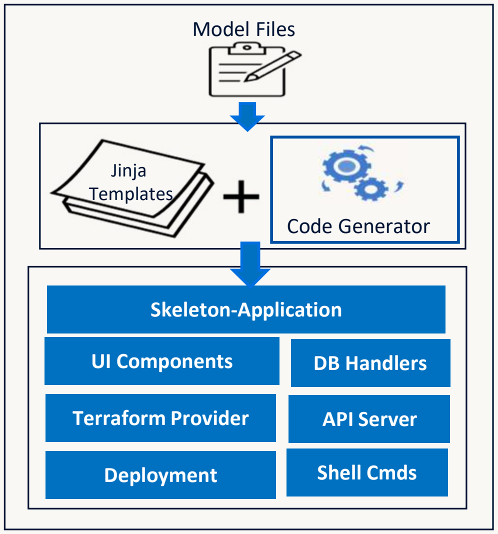
Dockerize

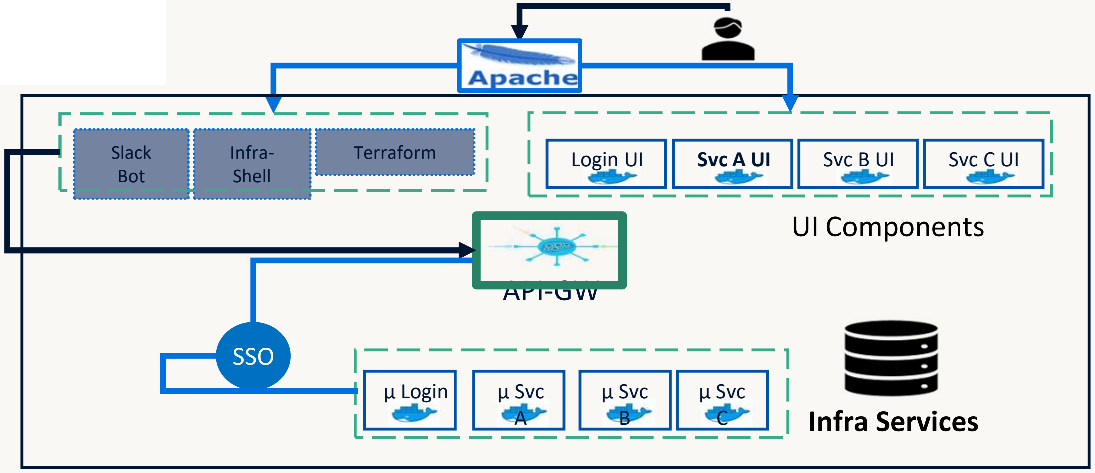
PayPal

Automating the automation

Framework to generate applications

- Goal was to automate all common tasks like UI coding, DB Handling etc
- Let the developers focus only on business logic
- Every application starts with a data model
- Jinja templates & Code generation engine gives out basic deployable application in minutes
- Generated application has code for DB, UI and other common tasks
- Standard Microservices architecture is the base for code generation
- New application ready for business logic development in minutes





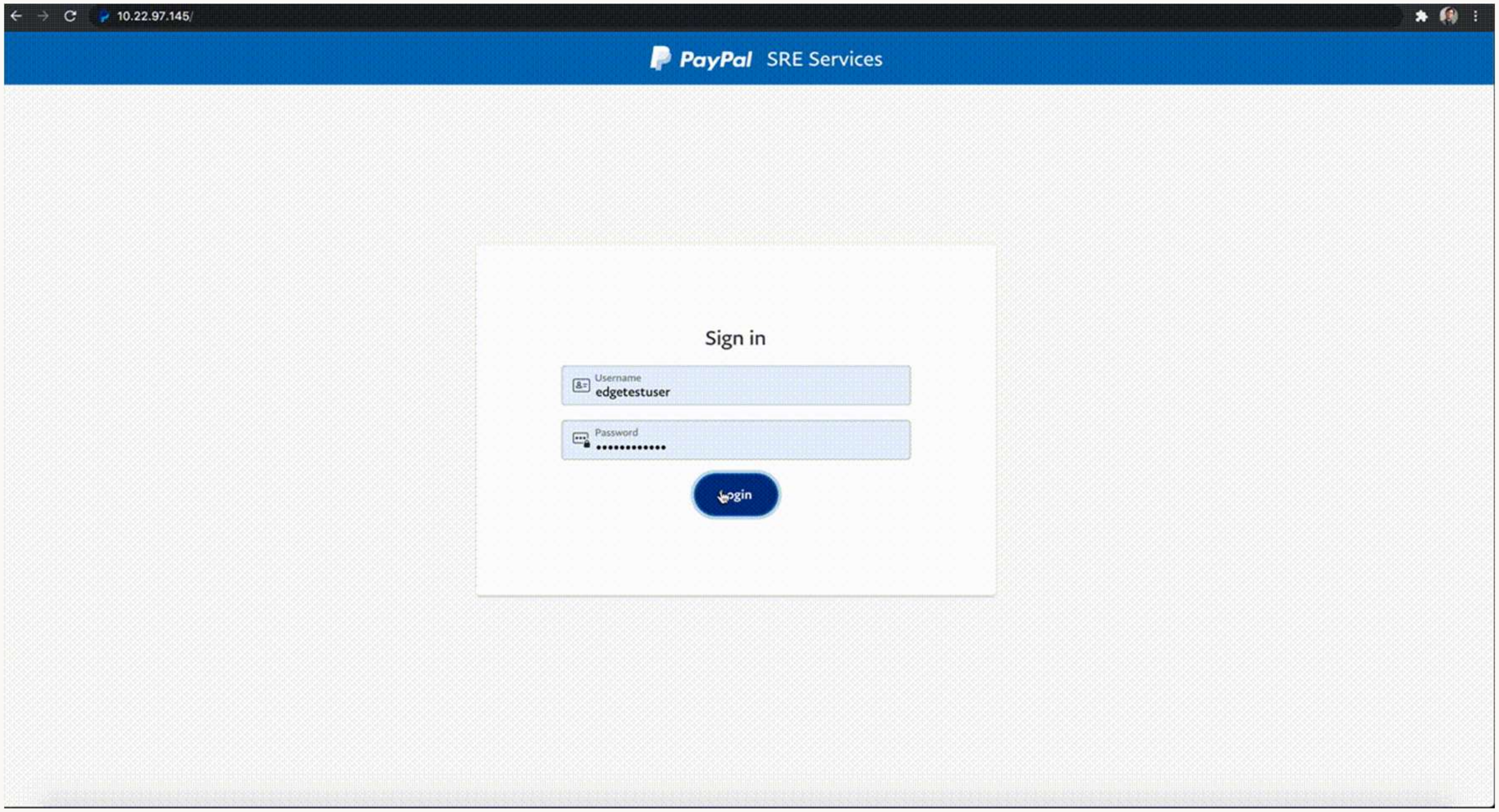
Application development in action

Steps to create a new service

- clone GNS2.0
- cd svcdefs
- mkdir <svc>
- cd <svc>
- touch <svc>_model.json
- Commit code
- Start Jenkins Build

```
songiri@LM-BGL-40518721 GNS2.0 %
```

Deployed application



Username
edgetestuser

Password
•••••••••••••

Results

- 25+ applications in 18months
- Uniform user interface across all applications
- Loved by command center teams, operation teams, and developer community
- Happy developers that can focus only on business logic
- Concept to release in record time for most applications
- Agile network with self service capabilities at every layer
- Decreased mean time to innocence, mean time to detection and mean time to resolution due to the new tools developed
- Expanding to rest of the domains within SRE

Closing thoughts…

- Cloud service providers have set new standards in infrastructure automation
- Scripts, Network automation does not meet the infrastructure consumers expectations
- Any internal automation efforts should look at the infrastructure as a platform
- Embrace microservices architectures
- Use proven opensource tools to your advantage
- Delivering more with less is the de-facto standard
- Produce services faster

Thank You

PayPal