

DevOps und Netzwerk für IPv6

Wie DevOps den Röstigraben
zwischen Systemen und Netzwerk
überwindet

Prinzipien Netzwerk

Prinzipien DevOps

Synthese (Agile Networking)

Agile Networking + IPv6

Ausgangslage

- Proprietäre Hardware
- Proprietäre Software
- Offene und proprietäre Protokolle

Struktur der Implementierung

- Horizontale Integration
- Schalenkonstruktionen (Core → Edge)
- IP-Layer kann Schalen quer schneiden oder auf Schalen lokal bleiben
- Zentrale Konfiguration durch zentrale Komponenten (z.B. eine zentrale Firewall)

Beispiel Unternehmensnetzwerk

- Core-Switche
- Gebäude-Switche
- Stockwerk-Switche

Beispiel Unternehmensnetzwerk

- Transfernetze ISP
- funktionale Netze (DMZ, Server, Clients, Management), organisatorische Netze (Abteilung A, B, C)
- Server-Netze oft näher/direkt am Core

Abstrakte Sicht

- Netzwerke managen Kanten eines Kommunikationsgraphen
- DevOps beschäftigt sich mit Blättern und setzt Graphen voraus

Change-Management

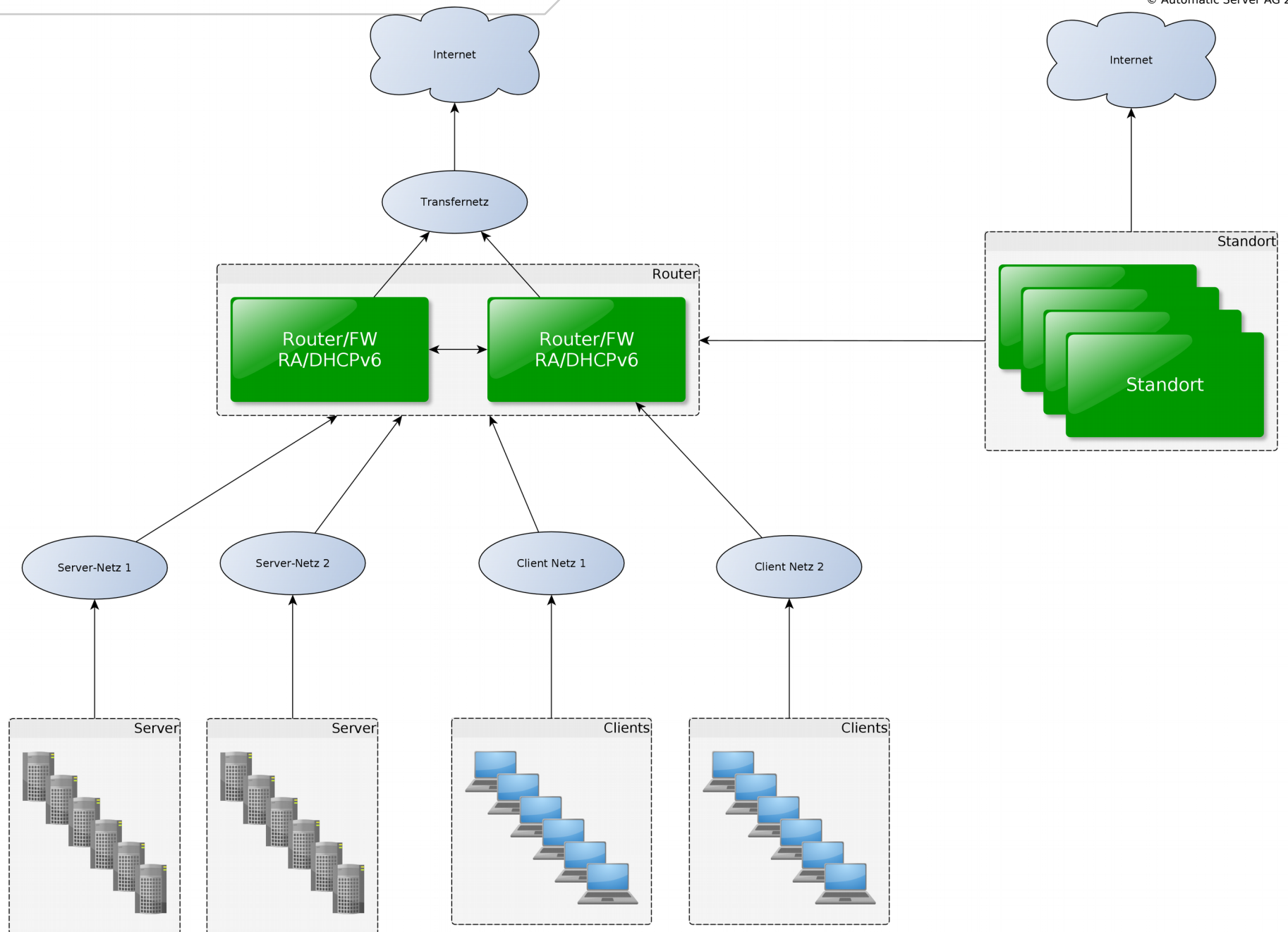
Was ändert sich?

- Zugangsregeln
- Firmware (Bugs, Sicherheitsprobleme, Features)
- Verfügbarkeit (selten da Produkt)
- Routing in grossen Netzen
- Konfigurations-Management hängt am Produkt

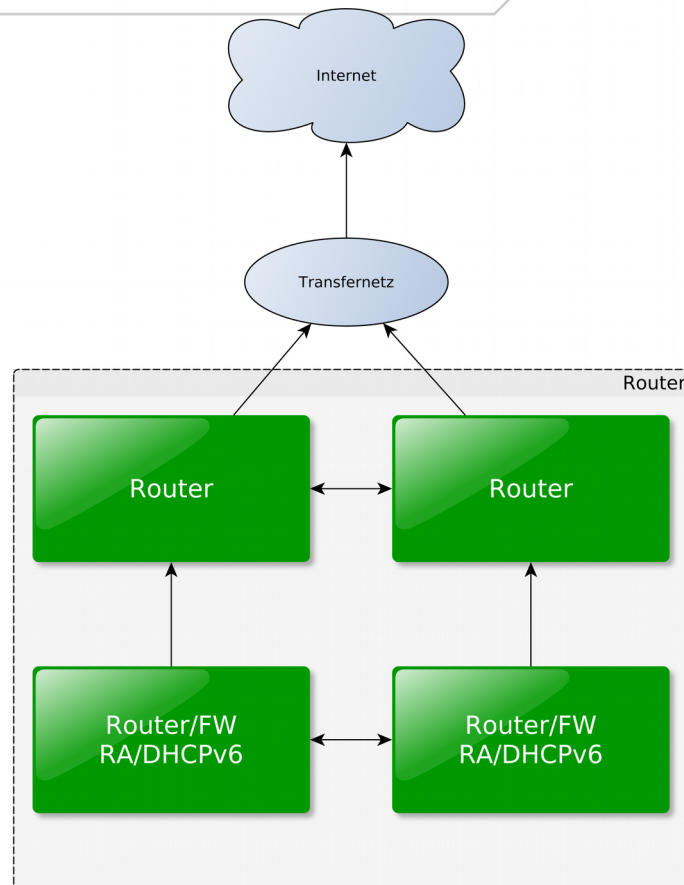
Tests

- Hardware kaufen
- Testaufbau (Kabel, Konfiguration)
- Test-Hosts anschliessen

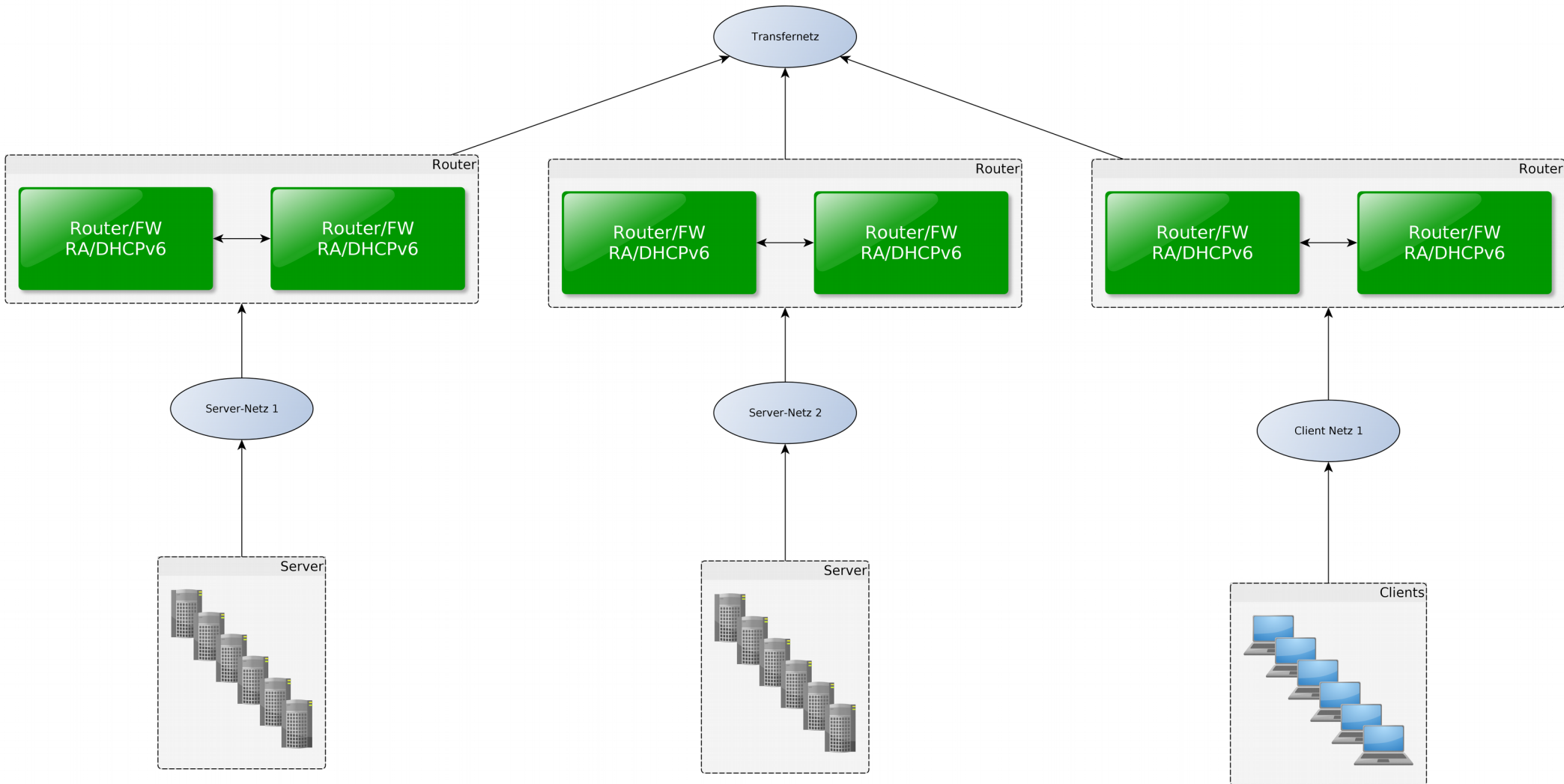
Prinzipien Netzwerk



Prinzipien Netzwerk



Prinzipien Netzwerk



Ausgangslage

- Offene Software
- Kostenlose Software
- Offene Schnittstellen
- Virtualisierung (Loslösung von Hardware)

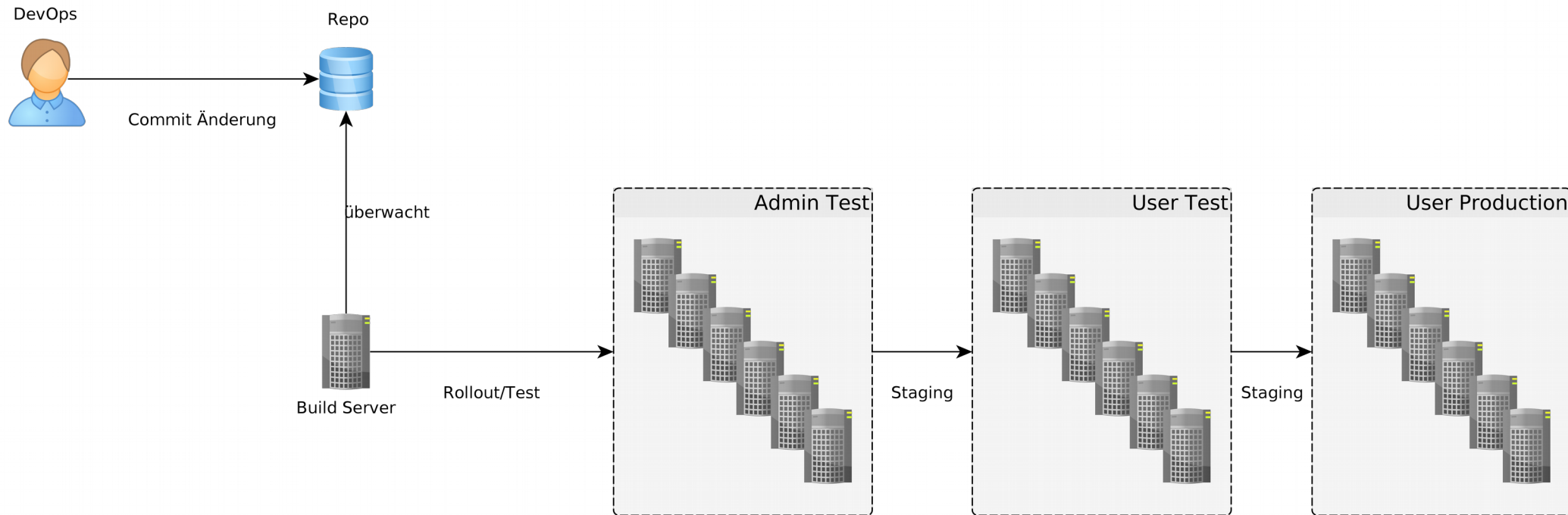
Struktur der Implementierung

- Möglichst grosse Isolation
- Testzentrierung und gestaffelte Testumgebungen
- Zentrale Konfiguration durch dezentrale Strukturen (z.B. Puppet + Parallel-SSH)

Ziel

So kurzfristig wie möglich in einer so hohen Qualität wie möglich reproduzierbar Änderungen herbeiführen um Software-Entwicklung optimal zu unterstützen

Prinzipien DevOps



Repository

Nachvollziehbarkeit

- Wer
- Wann
- Was

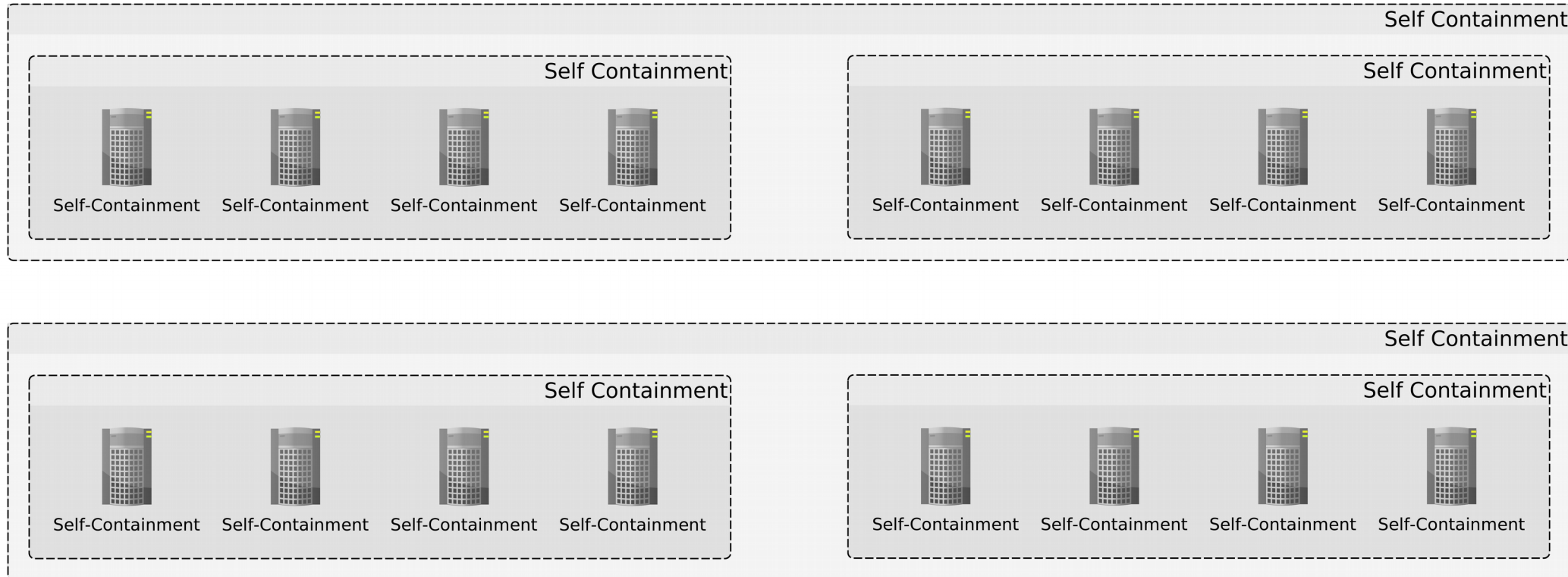
Repository

Historisierung gibt Rollback

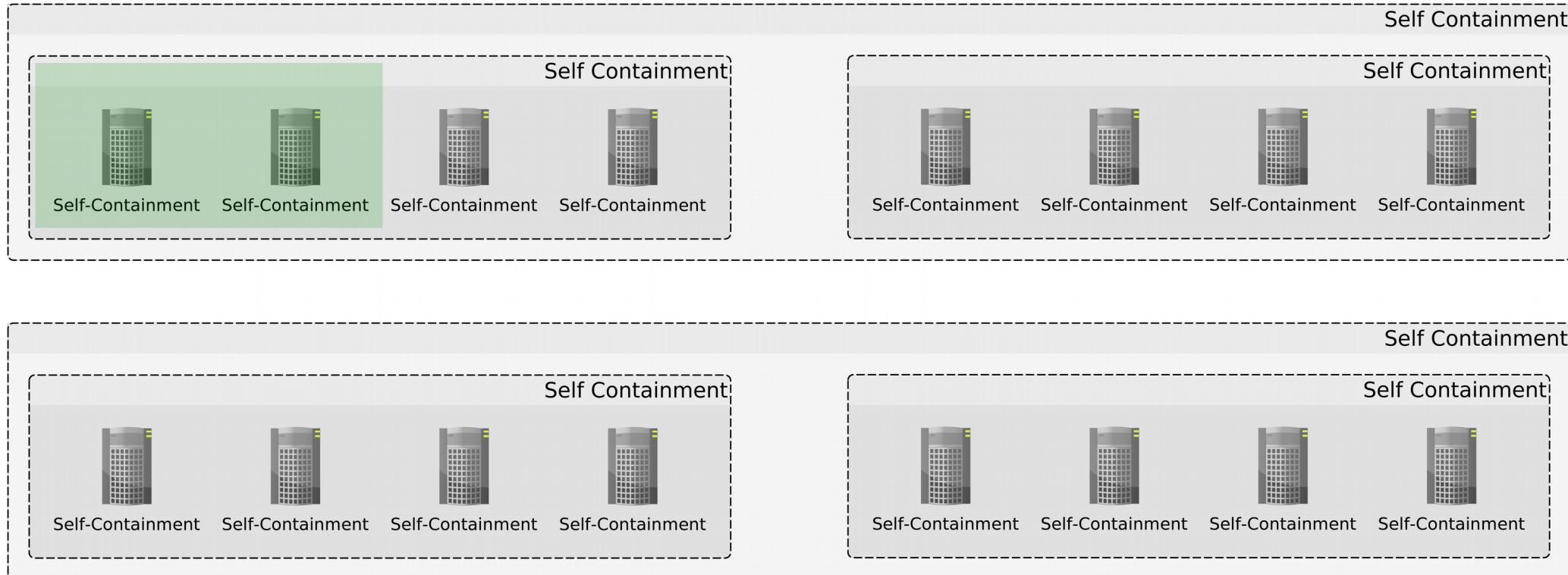
Build-Server

- Logging Ausrollprozess
- Automatische Tests
- Welche Änderung wurde wann wo ausgerollt
- Stellt gegebenenfalls Konfigurations-Dienst bereit wie z.B. Puppet

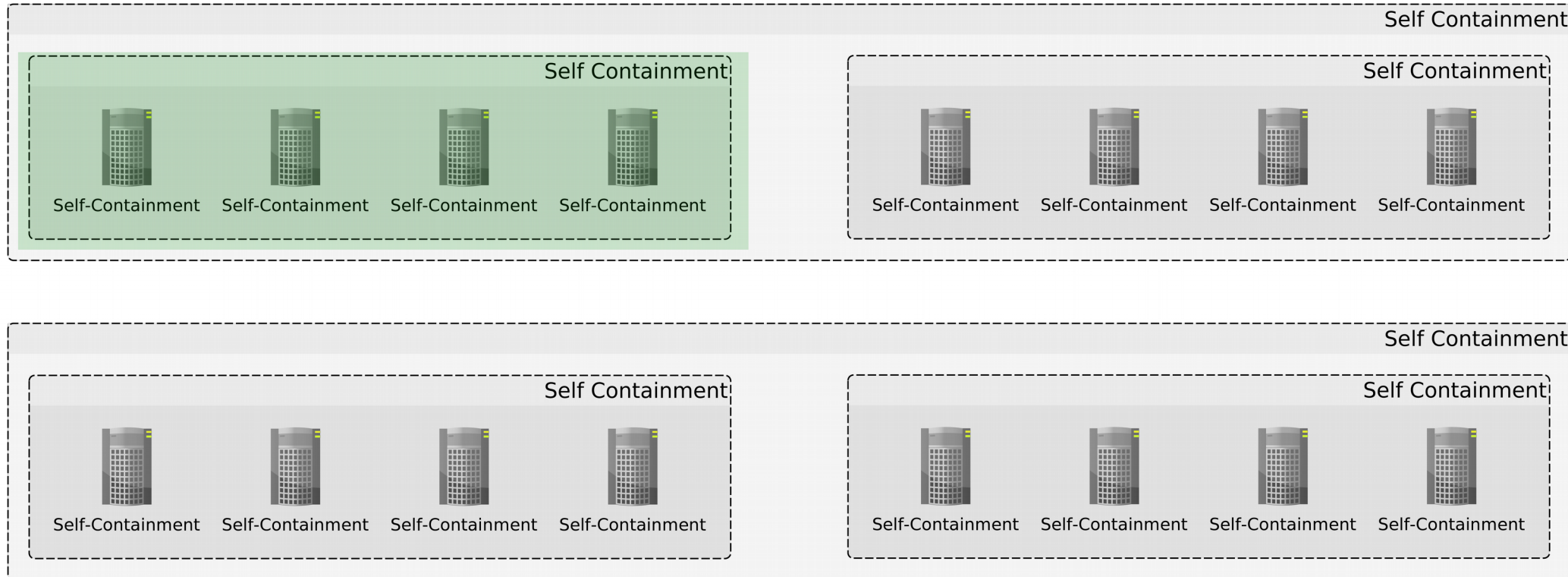
Prinzipien DevOps



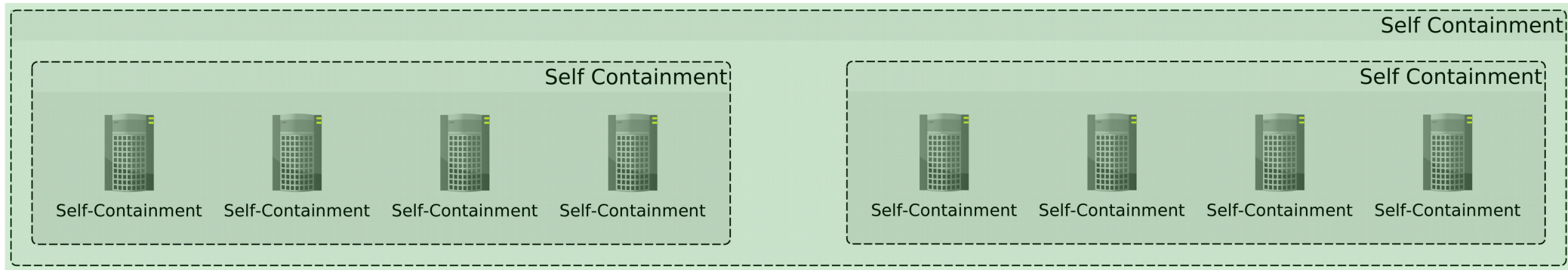
Prinzipien DevOps



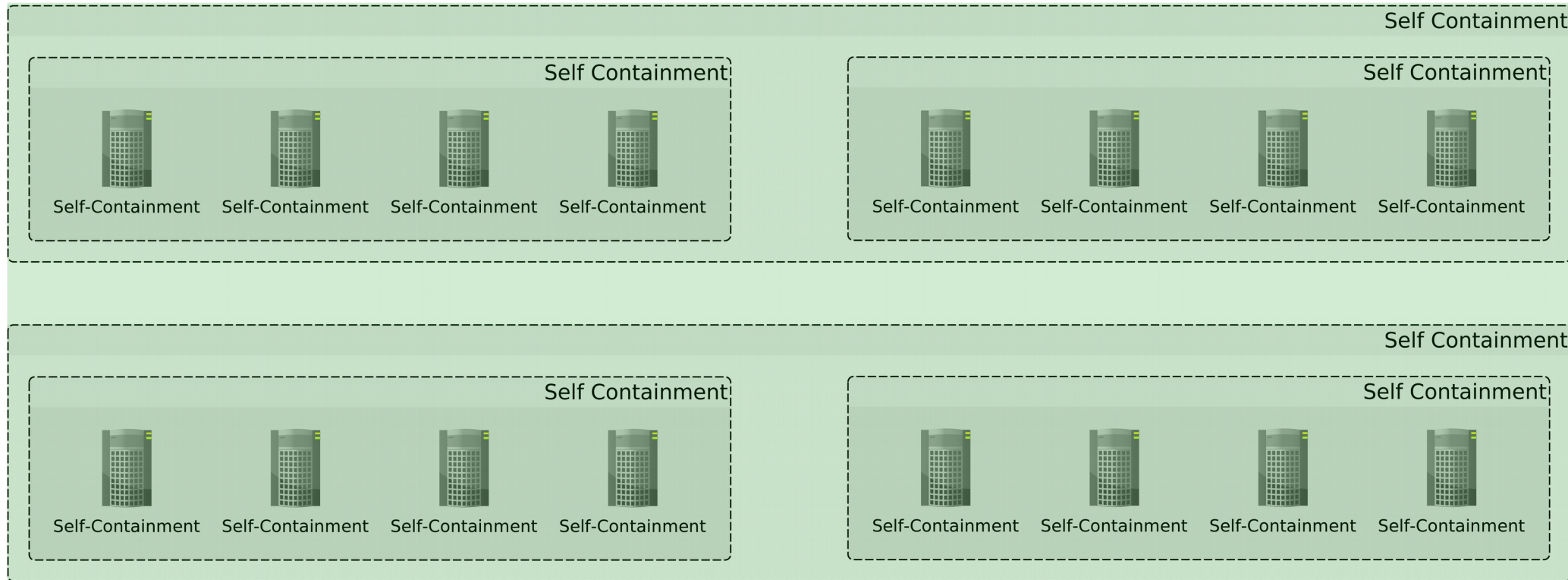
Prinzipien DevOps



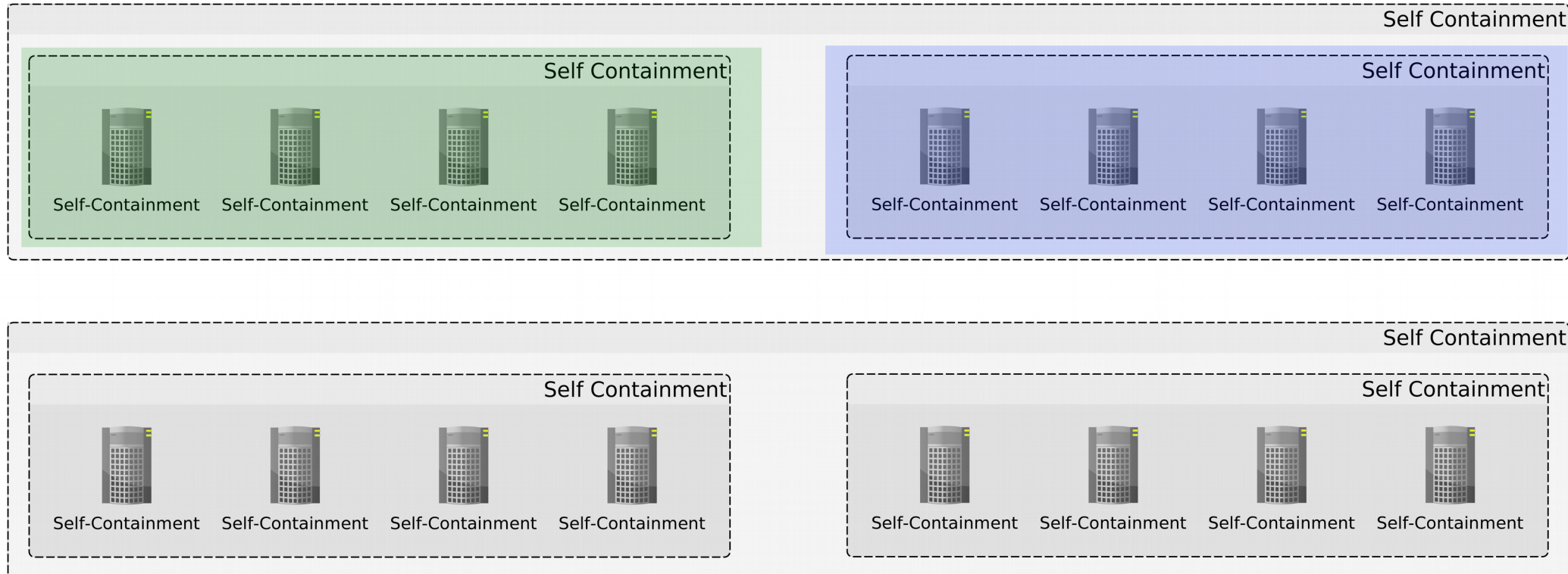
Prinzipien DevOps



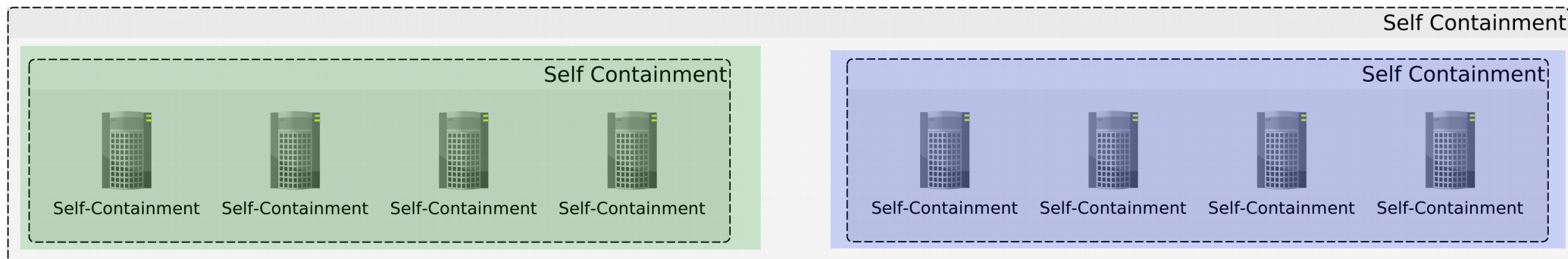
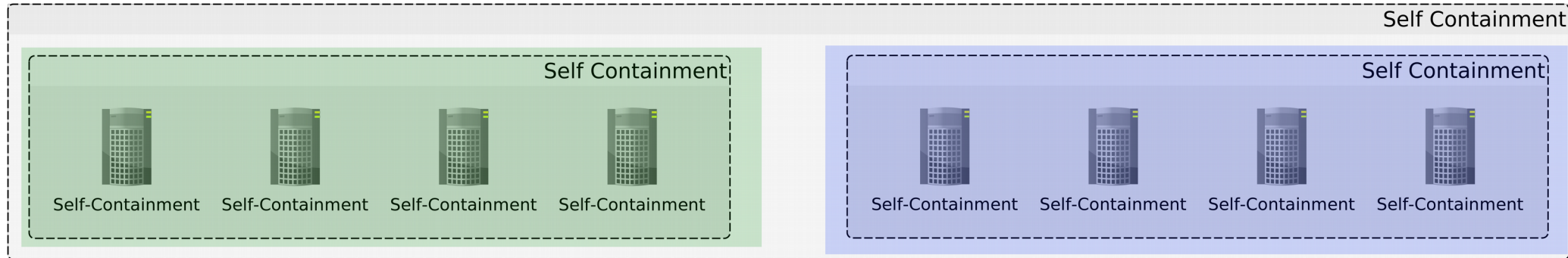
Prinzipien DevOps



Prinzipien DevOps



Prinzipien DevOps



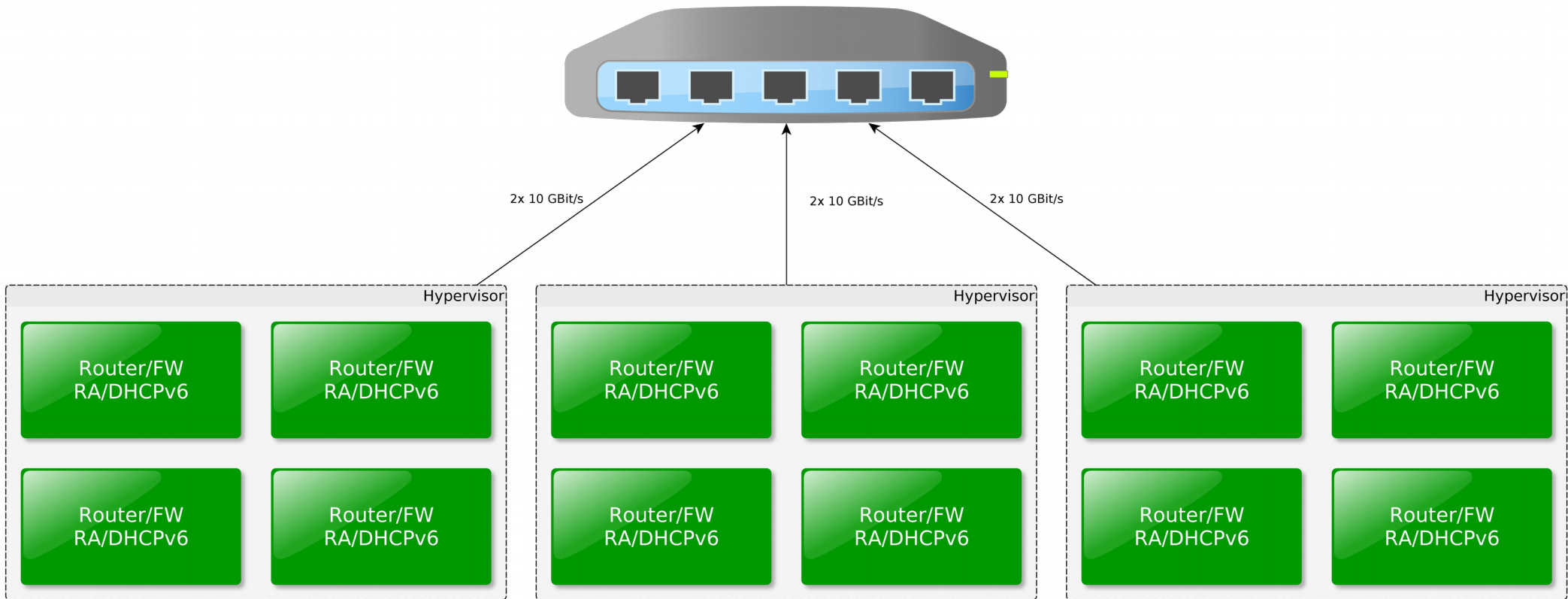
Self-Containment

- Testbarkeit herstellen
- Auswirkungen von Fehlern minimieren
- Konfigurations-Flexibilität
- Delegation von Verantwortung
- Verschiebbarkeit

Voraussetzungen

- Routing virtualisieren (SDN)
- Open-Source-Komponenten verwenden
- Netzwerk vertikalisieren
- Funktionstests etablieren

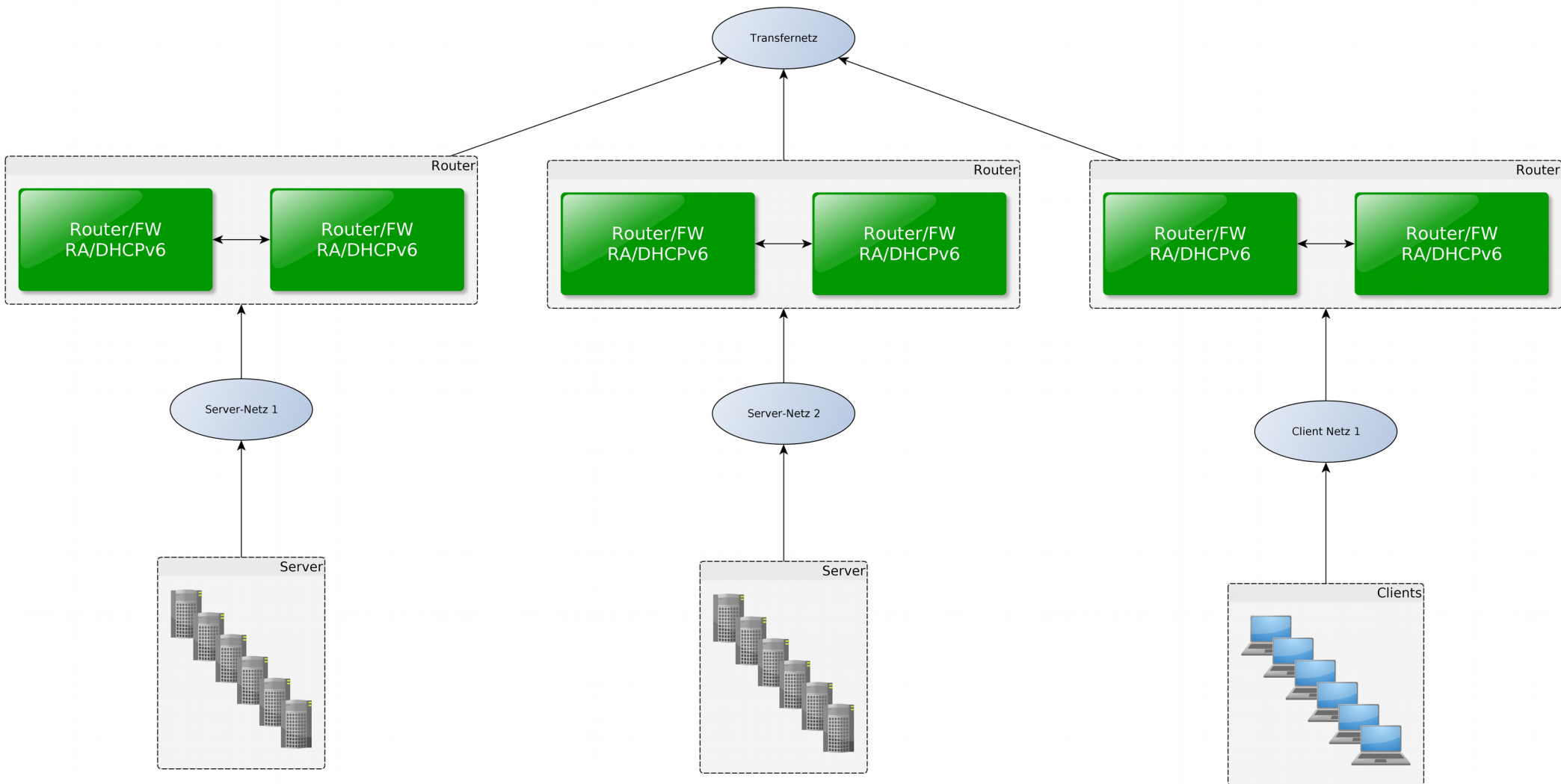
Agile Networking



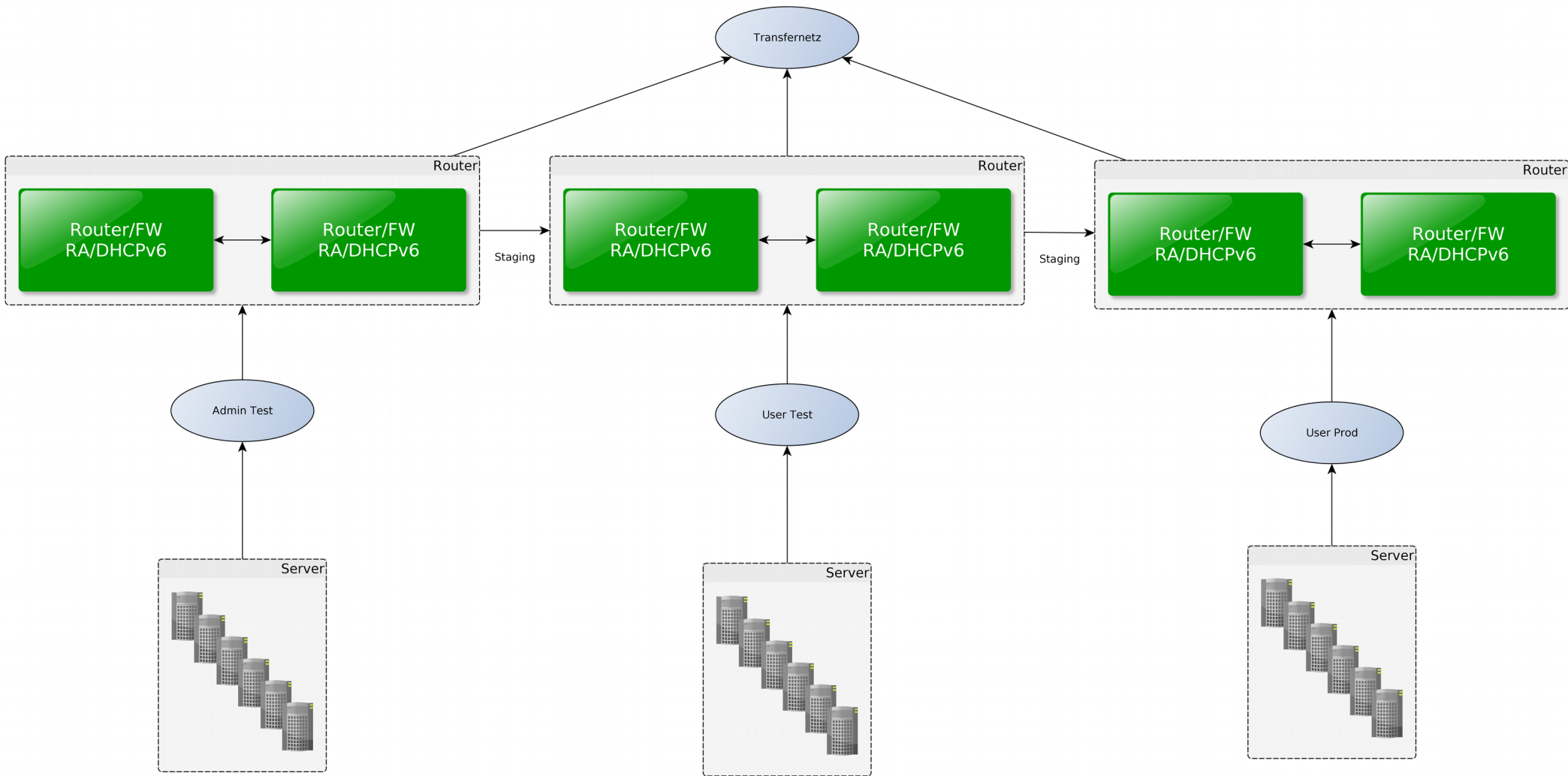
Virtuelle Router

- BSD Router Project, FreeBSD, NetBSD
- Linux (VyOS/Vyatta), Debian, Ubuntu, ...

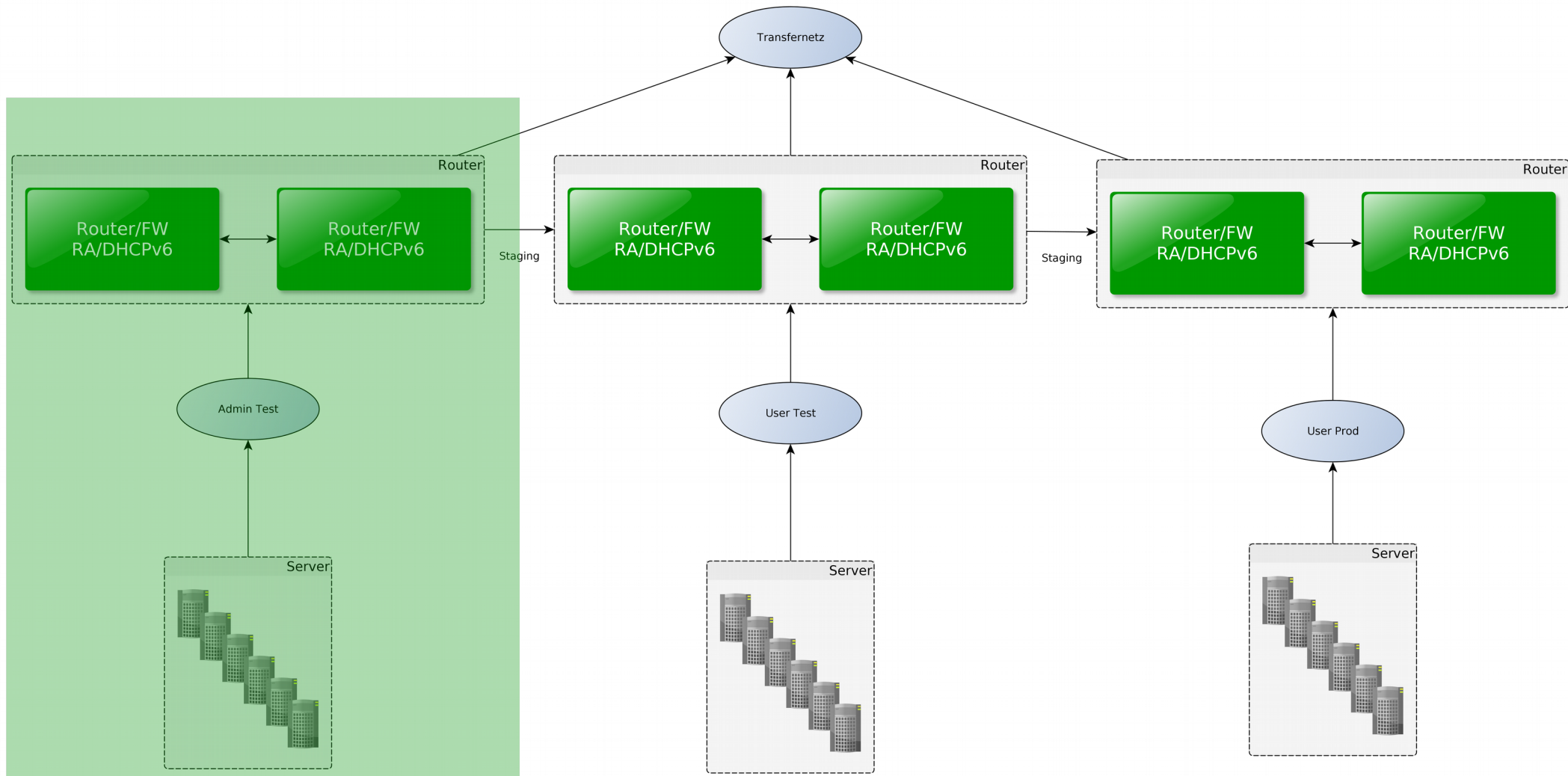
Prinzipien Netzwerk



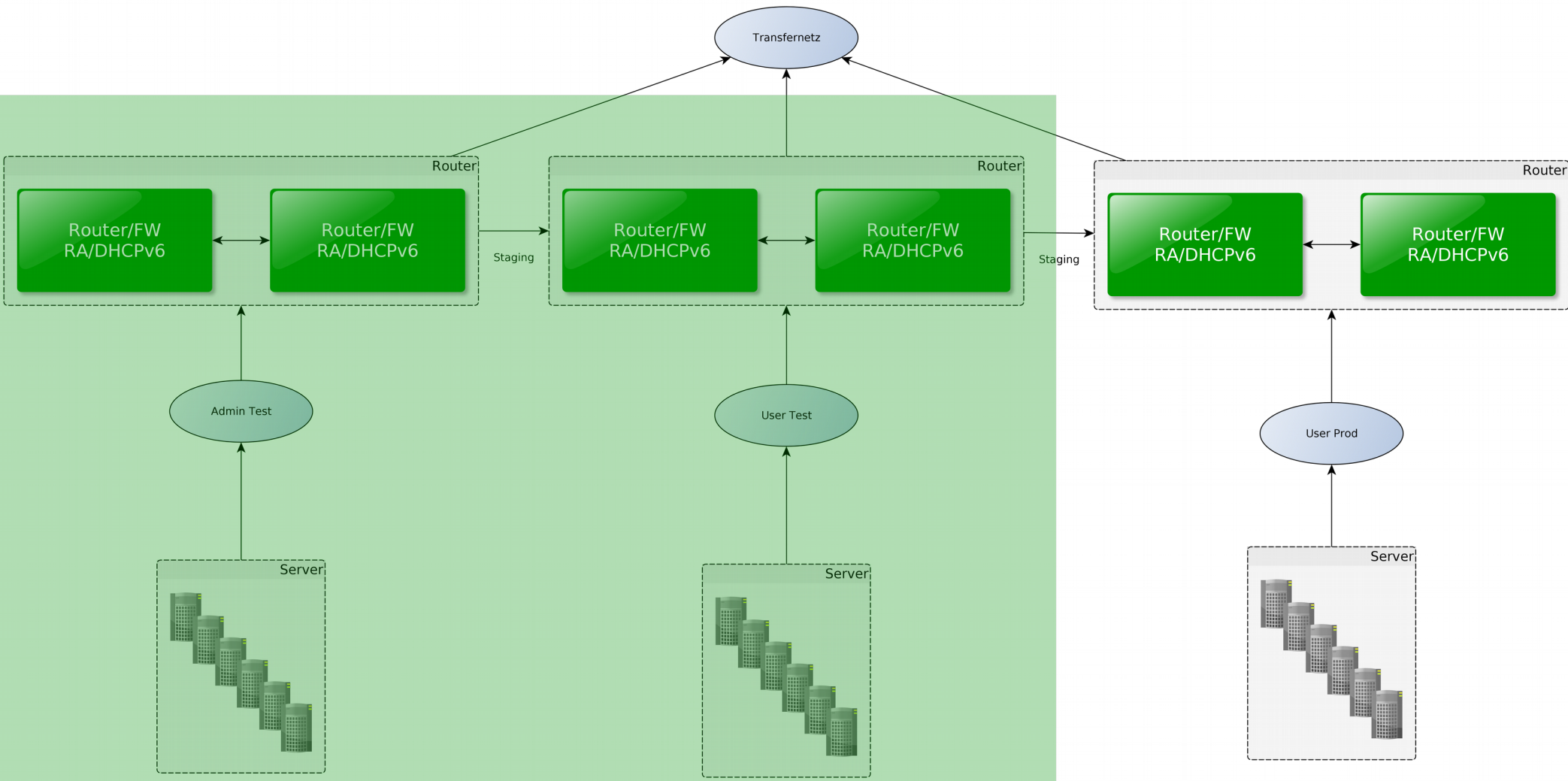
Agile Networking



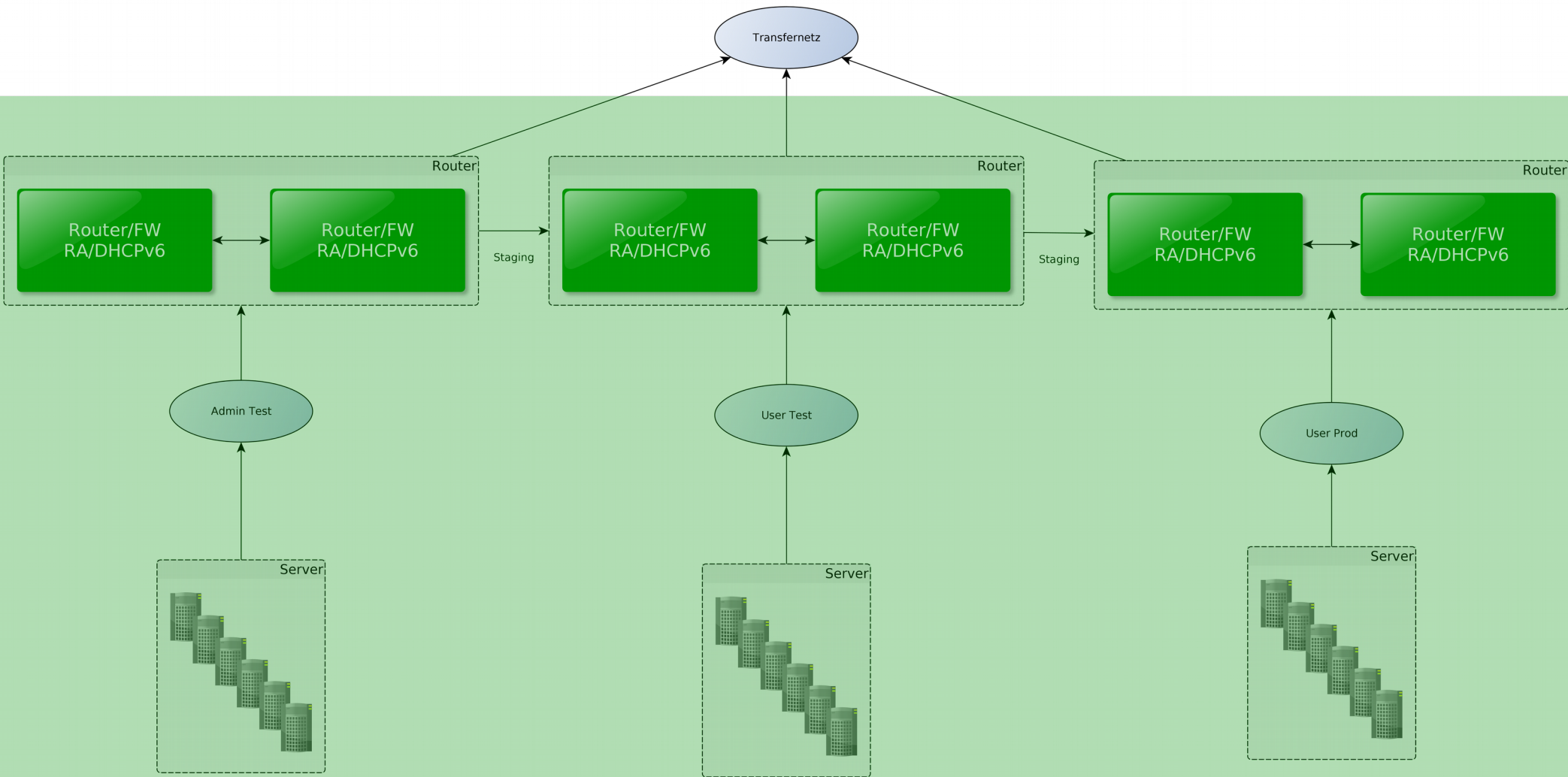
Agile Networking



Agile Networking



Agile Networking



Testen

- Kann man einen Test formulieren, der Konnektivität prüft?
- Kann ein Kommunikationsgraph getestet werden?
- Kann ein Referenz/Test-Kommunikationsgraph erstellt werden?
- Kann eine eineindeutige Abbildung von Referenz- auf Produktionsgraph vorgenommen werden?
- Kann man einen Vergleich automatisch analysieren zwischen zwei Tests?
- Kann man höherwertige Tests (Server/Applikation) rufen?
- Kann man einen Test/Build-Server netzlokal mit diesen Tests beauftragen?
- Können Referenz-Hosts oder alle Hosts getestet werden?

Anwendungsfälle

- Automatisches Rollout Sicherheits-Updates Router/Firewalls
- Automatisches Rollout Konfigurationseinstellungen
- Rollout IPv6

2 Phasen

- Netzwerk
- Hosts (Adressvergabe) + Applikationstests

Rollout-Planung Netzwerk

- Adresskonzept erstellen
- Testkonzept erarbeiten
- Risikonetze ermitteln
- Tests schreiben (programmieren/konfigurieren)
- IPv6-Testzone erstellen
- manuelle IPv6-Tests in IPv6-Testzone
- Adress- und Routen-Rollout in Netzwerkinfrastruktur inkl. Ausführung Tests zonenweise

Rollout-Planung Hosts + Applikation

- Referenz-Hosts ermitteln (Hostklassen)
- Referenz-Applikationen ermitteln
- Risiko-Hosts ermitteln
- Risiko-Applikationen ermitteln
- Test-Entwicklung auf Host-Ebene und Aufruf aus Netzwerk-Build-Pipeline ermöglichen
- Test-Entwicklung auf Applikationsebene und Aufruf aus Netzwerk-Build-Pipeline ermöglichen

Risikomanagement

Risiko-Netzwerke, -Hosts oder -Applikationen werden nur halbautomatisch im Rollout einbezogen, d.h. die Prüfung enthält eine manuelle Abnahme

Klassiker:

- Drucker, IP-Telefone
- WLAN
- Alte Soft- und Hardware
- Proxies/ALGs