



**Connect**

# Edge-App-Deployments vereinfachen – ein Workflow für Tausende Geräte

**Adrian Luberdá**

[aluberdá@redhat.com](mailto:aluberdá@redhat.com)

Solution Architect

Red Hat

**Christian Koep**

[ckoep@redhat.com](mailto:ckoep@redhat.com)

Solution Architect

Red Hat





## Edge Computing?

Rechenkapazität nahe an der Datenquelle

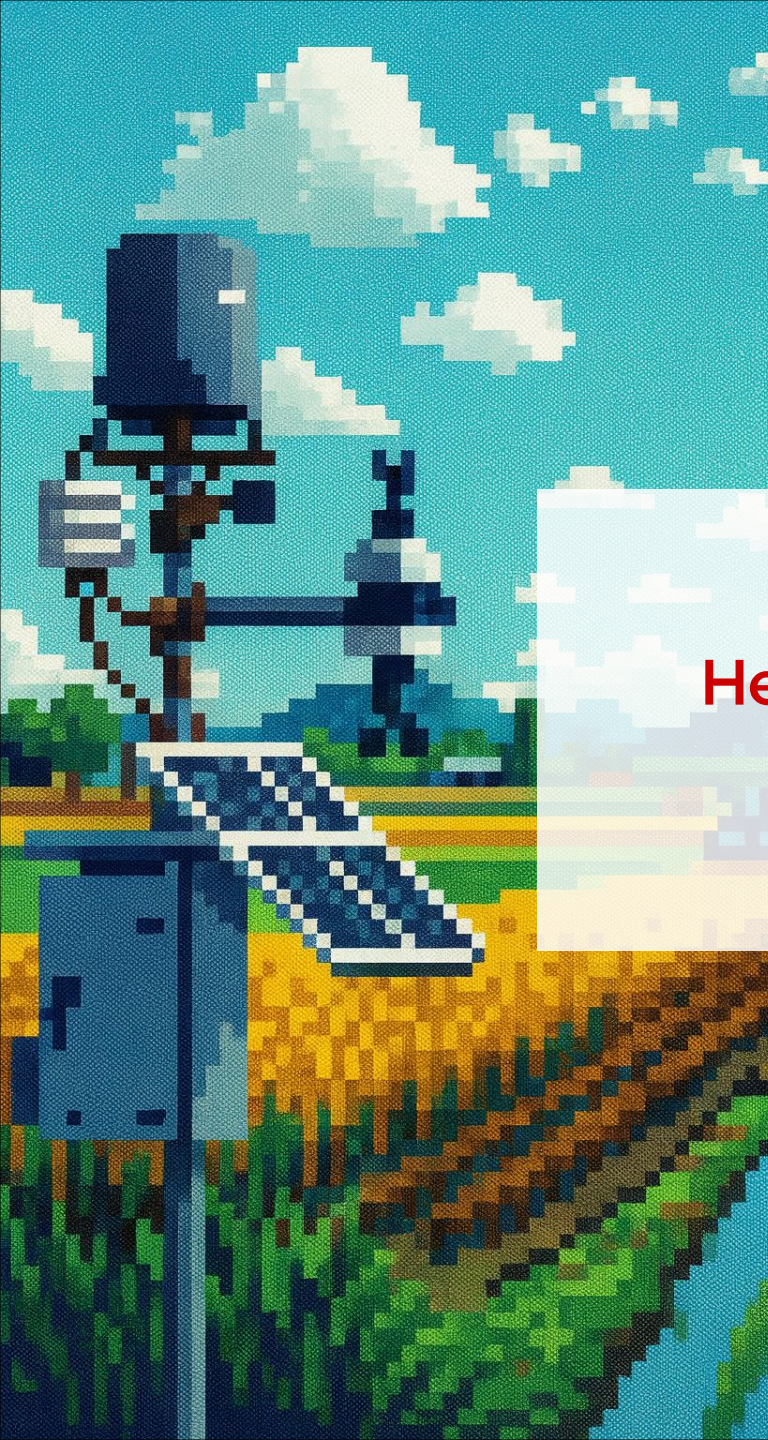
Schnellere Reaktion nahe am Asset

Entscheidungen mit geringer Latenz

Hohe Resilienz durch lokale Ausführung



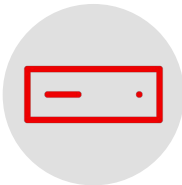




## Herausforderungen beim Aufbau von Edge-Computing-Plattformen



# Herausforderungen: Ressourcen



## Begrenzte Hardware

Hardware in Edge-Geräten ist für einen dedizierten Anwendungsfall ausgelegt



## Netzwerkanbindung

Fragmentierte Netzwerkstabilität und limitierte Bandbreite



## Vor-Ort Personal

An abgelegenen Standorten fehlt es häufig an IT-Personal für Wartung und Fehlerbehebung



# Herausforderungen: Stabilität und Sicherheit



## Konfiguration

Systemzustände driften mit der Zeit auseinander, wodurch Inkonsistenzen entstehen.



## Updates

Update sind riskant, da Rollbacks oft nicht möglich sind. meistens fehlen.



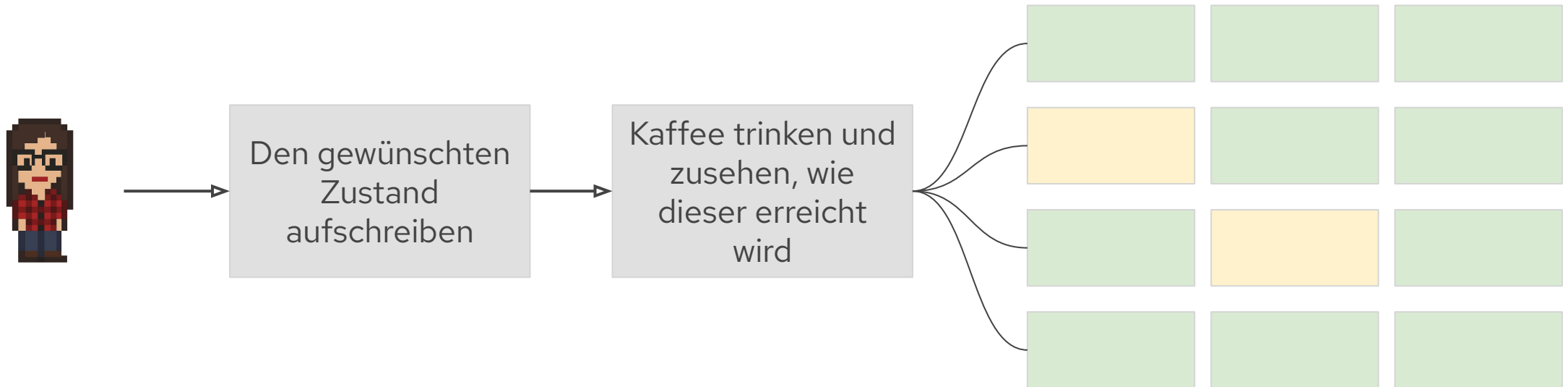
## Software Supply Chain

Die Absicherung des Codes von der Entwicklung bis zum Endgerät passiert auf Best-Effort

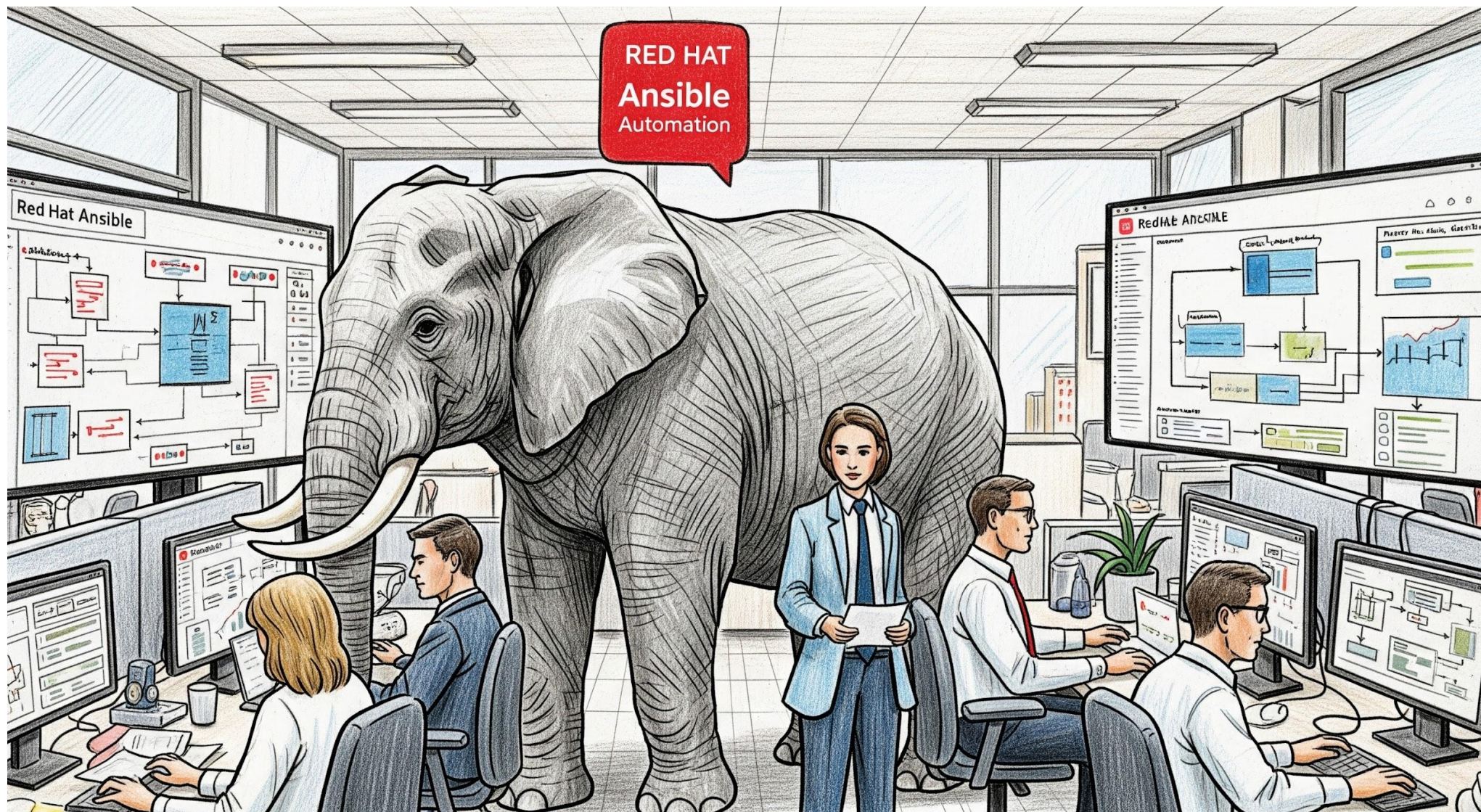


# Lösung: Ein simples Verfahren zum Deployment von Edge-Anwendungen

Was wäre, wenn das Deployment auf Tausenden Geräten so einfach wäre wie auf einem einzigen?







RED HAT  
Ansible  
Automation

Red Hat Ansible

Red Hat Ansible



Heute sprechen wir über

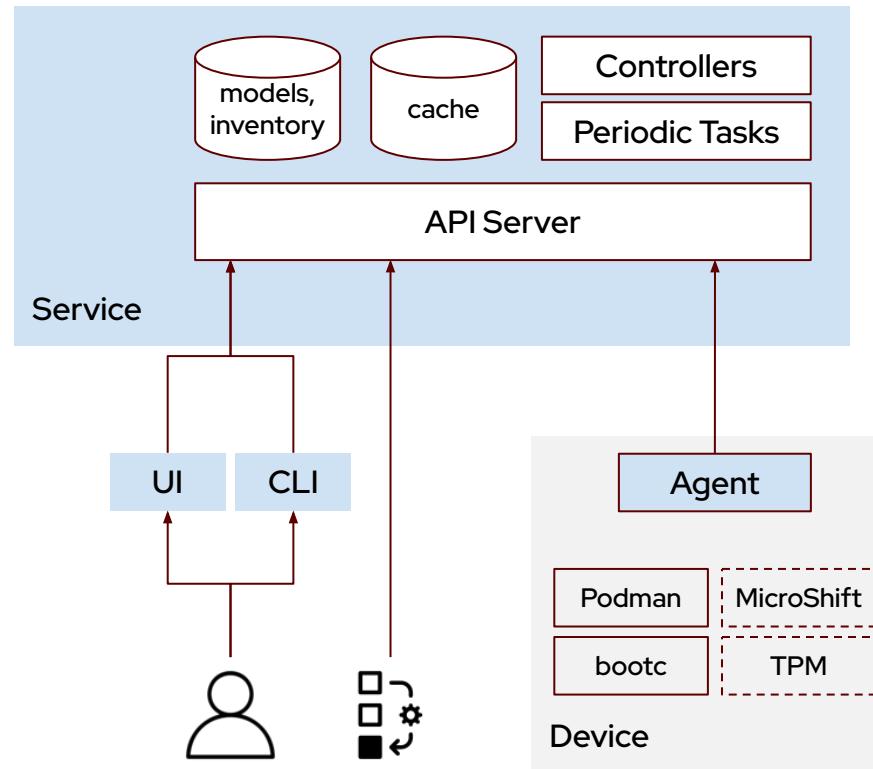


als Ergänzung zu Ansible!





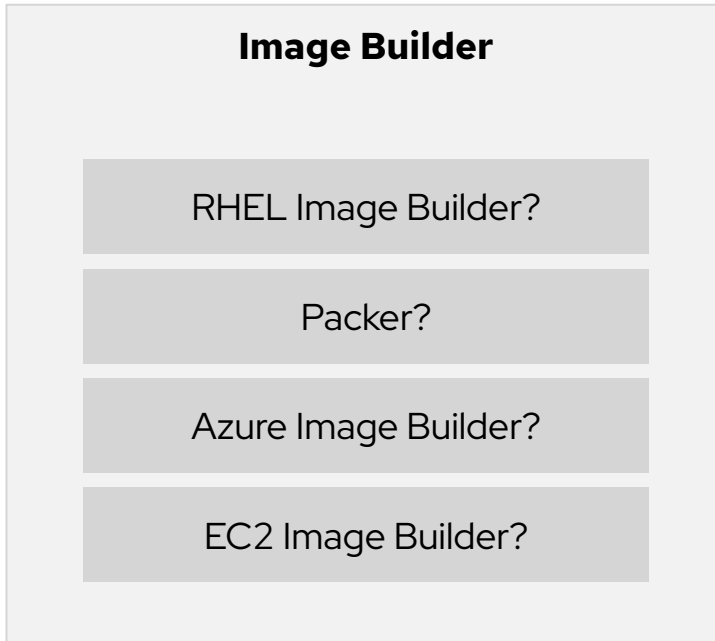
# Agentenbasierte Architektur



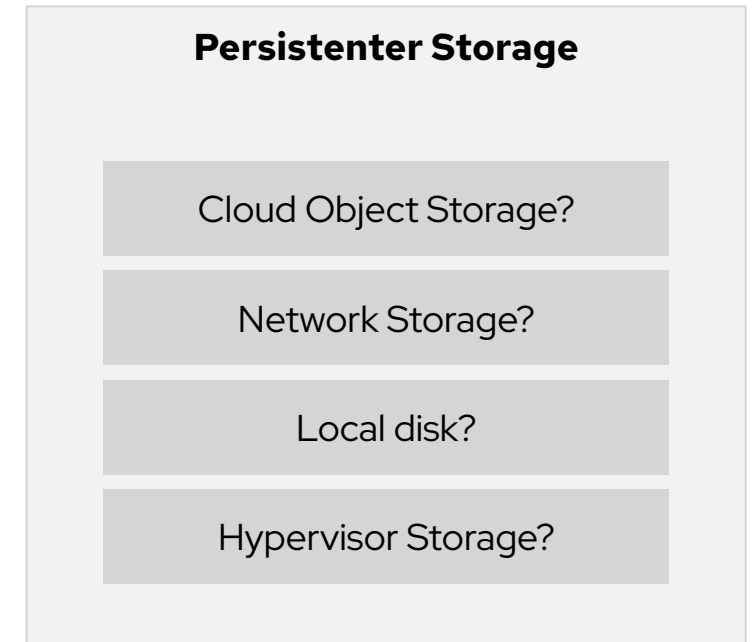
Ein Blick hinter die  
Kulissen: **bootc**



# Herausforderung: Linux Images bauen



.qcow2, .vhd, .ami, .iso, ...



## Direkter Vergleich: Container Images bauen







Was wäre, wenn es möglich wäre, beide Ansätze  
zu kombinieren?

# Lösung: Ein Tool für Linux- und Container Images







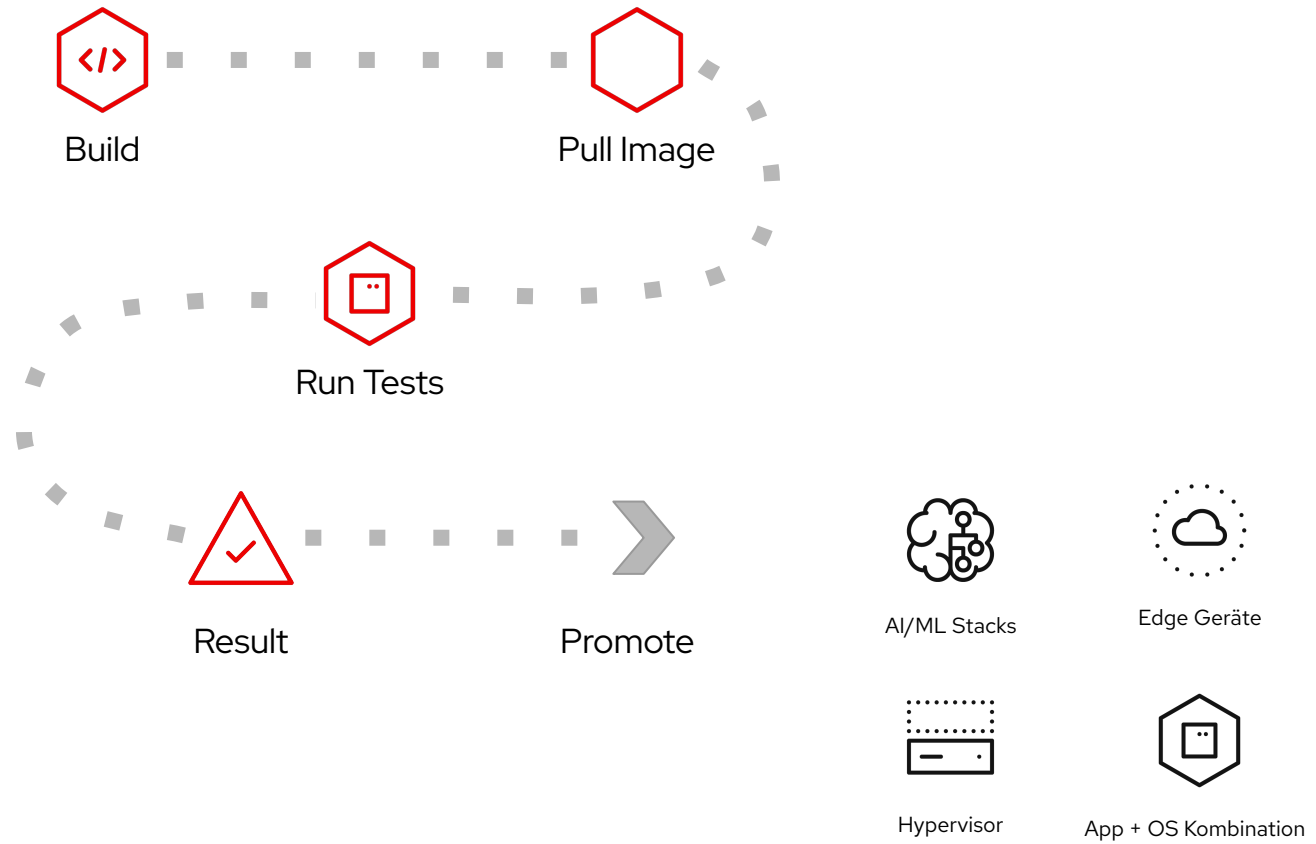
```
FROM registry.redhat.io/rhel10/rhel-bootc:10.0
RUN dnf install -y httpd && \
    dnf clean all

RUN systemctl enable httpd.service

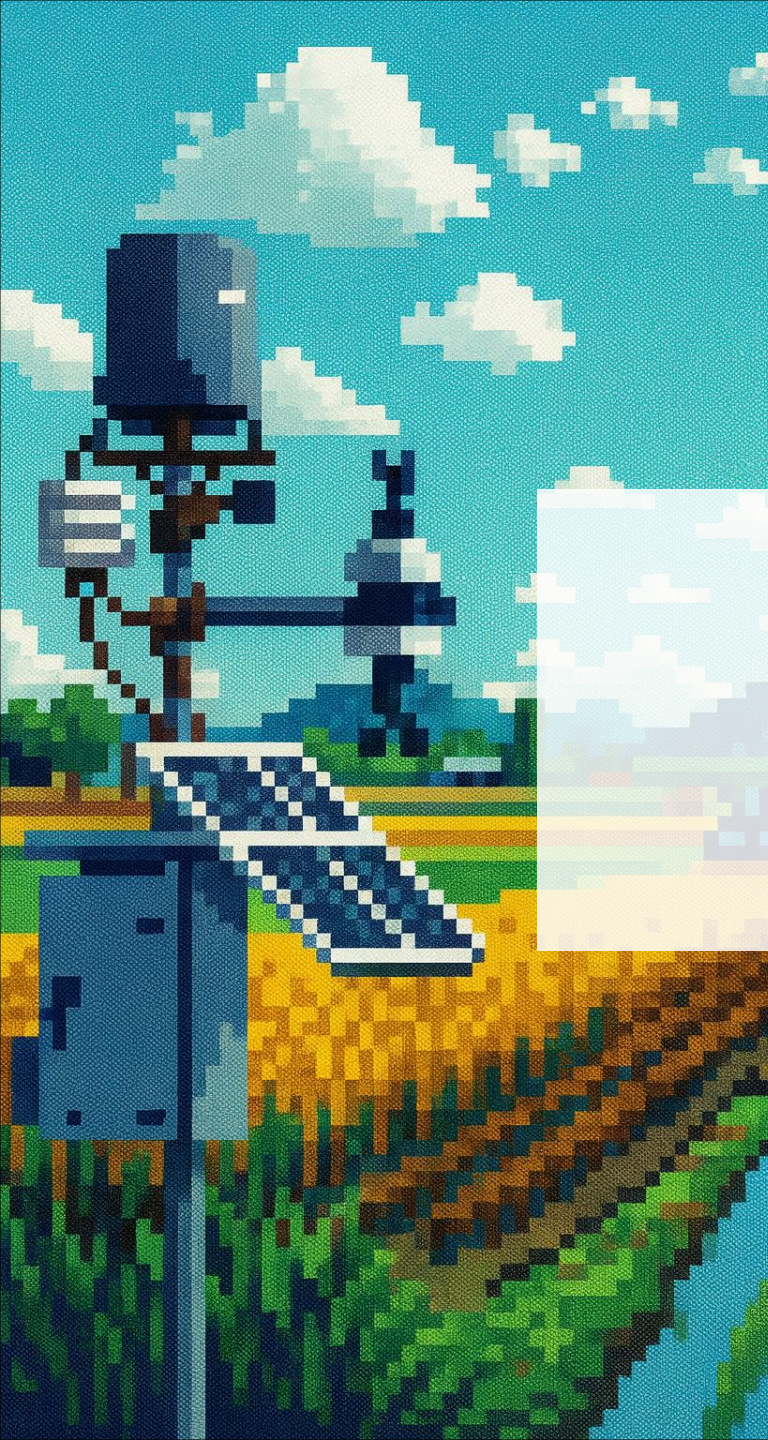
COPY index.html /var/www/html/index.html
```



# Die Evolution von Runtime zu Buildtime Konfiguration







## Herausforderungen mit Edge-App-Deployments





# Herausforderungen mit Edge-App-Deployments



## **Ende-zu-Ende Sicherheit**

Von der Erstinstallation über den Aufbau an einer Edge-Location bis hin zu Aktualisierungen.



# Lösung: Sichere Installation und Software-Deployments

Zero-Touch und Zero-Trust durch den FIDO Device Onboard (FDO) Standard



 Digital signiert und überprüft

 Orchestriert über mTLS



# Lösung: Sichere Installation und Software-Deployments

Signierte Anwendungen, Betriebssysteme und TLS-Zertifikate

**Kompromittiertes Edge Device**



🔒❌ Nicht-signierte Anwendung startet nicht

🔒❌ Keine Verbindung zum Red Hat Edge Manager





## Zusammengefasst: Wie ist der Prozess, um ein neues Gerät auszurollen?



Techniker ist vor Ort  
Schaltet das Gerät ein  
Stellt die Netzwerkverbindung her  
Gibt das Gerät frei



**DEMO**

# Herausforderungen mit Edge-App-Deployments



## Ende-zu-Ende Sicherheit

Von der Erstinstallation über den Aufbau an einer Edge-Location bis hin zu Aktualisierungen.



## Konfigurationsabweichungen

Es muss sichergestellt werden, dass sämtliche Endgeräte dieselbe Konfiguration aufweisen - dauerhaft.





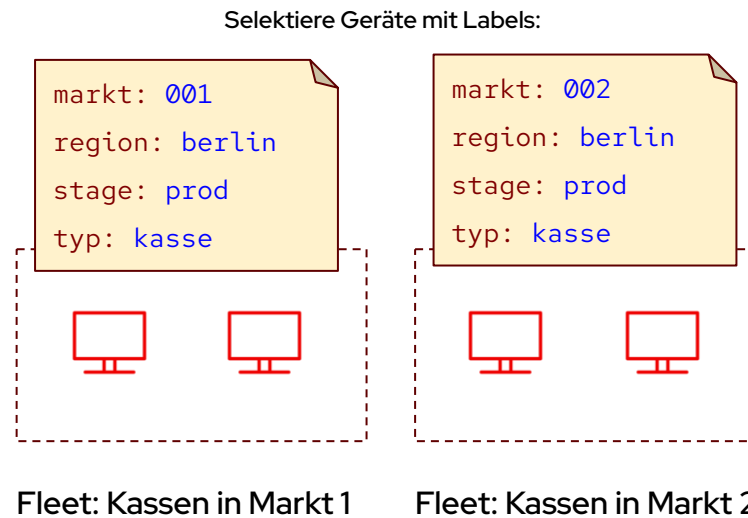
# Herausforderung: Konfigurationsabweichungen

**Soll-Zustand**



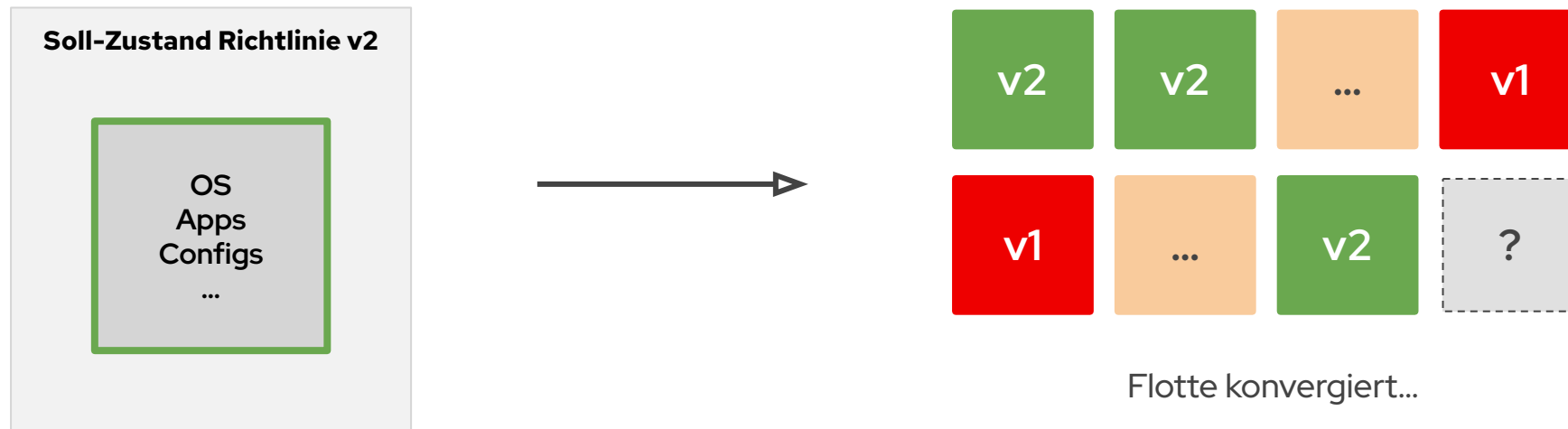
# Lösung: Deklarative Konfiguration und Abstraktion

“Fleets” definieren Soll-Zustand, Labels ordnen Geräte zu



## Lösung: Konsistenz durch Richtlinien

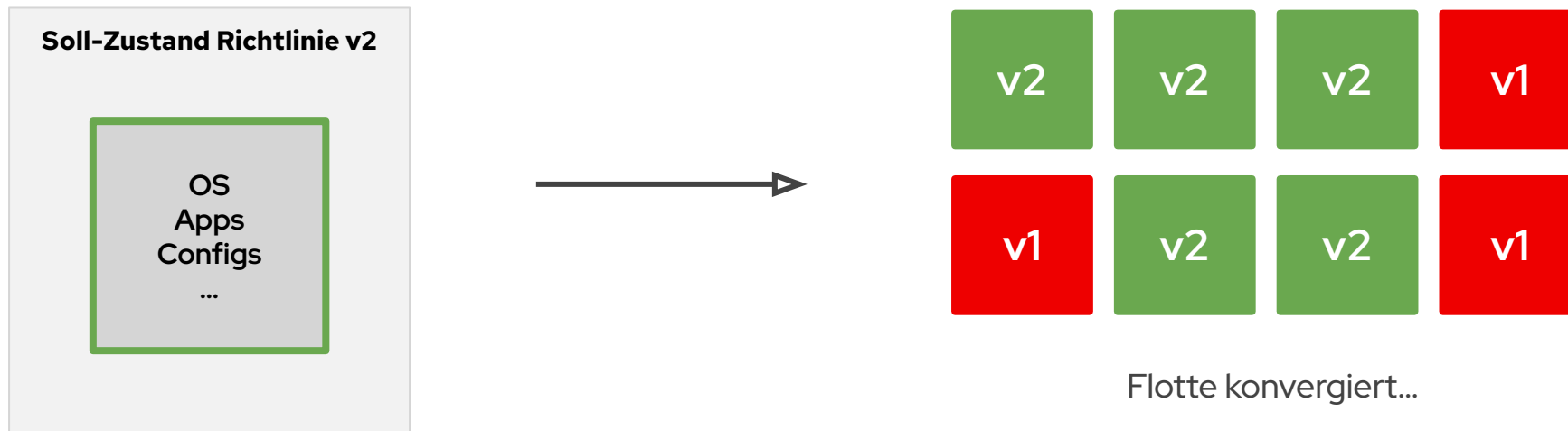
Automatische Angleichung von Soll- und Ist-Zustand





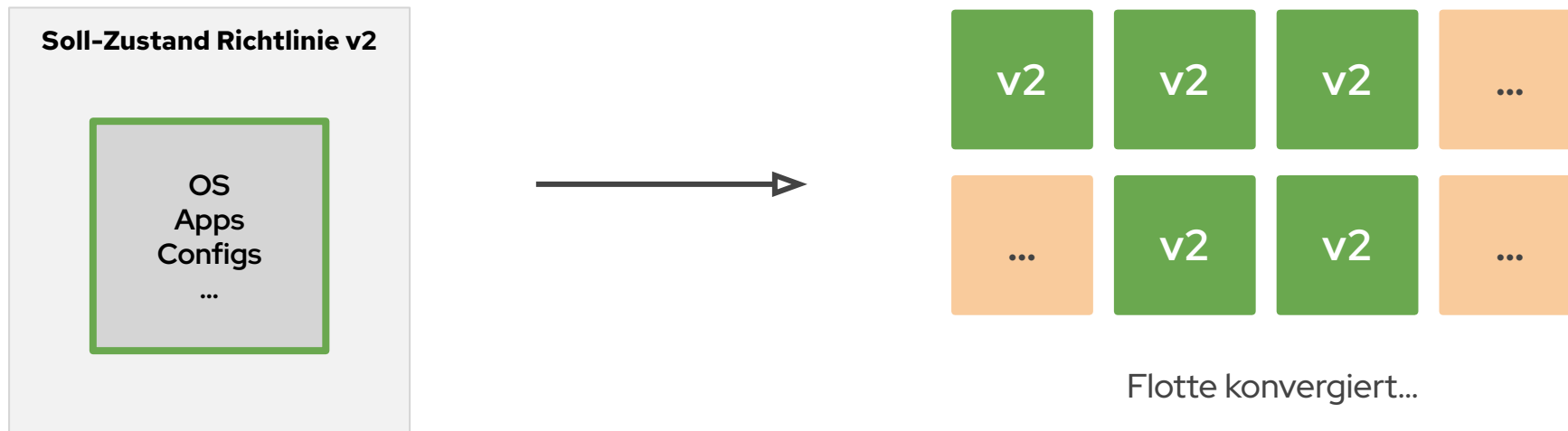
## Lösung: Konsistenz durch Richtlinien

Automatische Angleichung von Soll- und Ist-Zustand



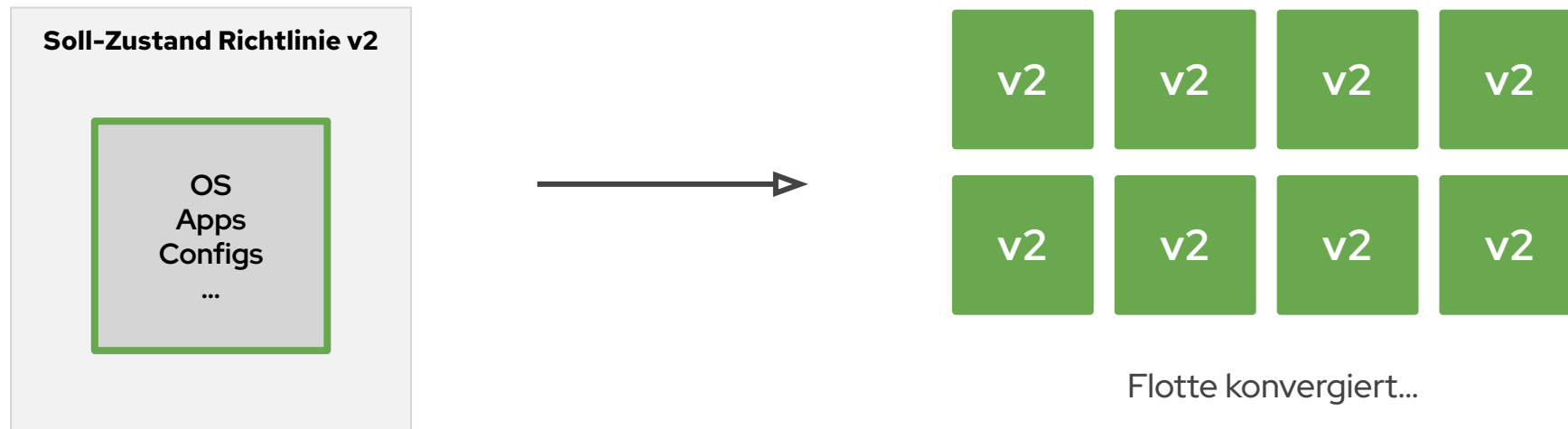
## Lösung: Konsistenz durch Richtlinien

Automatische Angleichung von Soll- und Ist-Zustand



## Lösung: Konsistenz durch Richtlinien

Automatische Angleichung von Soll- und Ist-Zustand



**DEMO**



# Herausforderungen mit Edge-App-Deployments



## Ende-zu-Ende Sicherheit

Von der Erstinstallation über den Aufbau an einer Edge-Location bis hin zu Aktualisierungen.



## Konfigurationsabweichungen

Es muss sichergestellt werden, dass sämtliche Endgeräte dieselbe Konfiguration aufweisen - dauerhaft.



## Troubleshooting

Kein Personal vor Ort oder Personal mit wenig IT-Erfahrung oder adäquaten Werkzeugen.



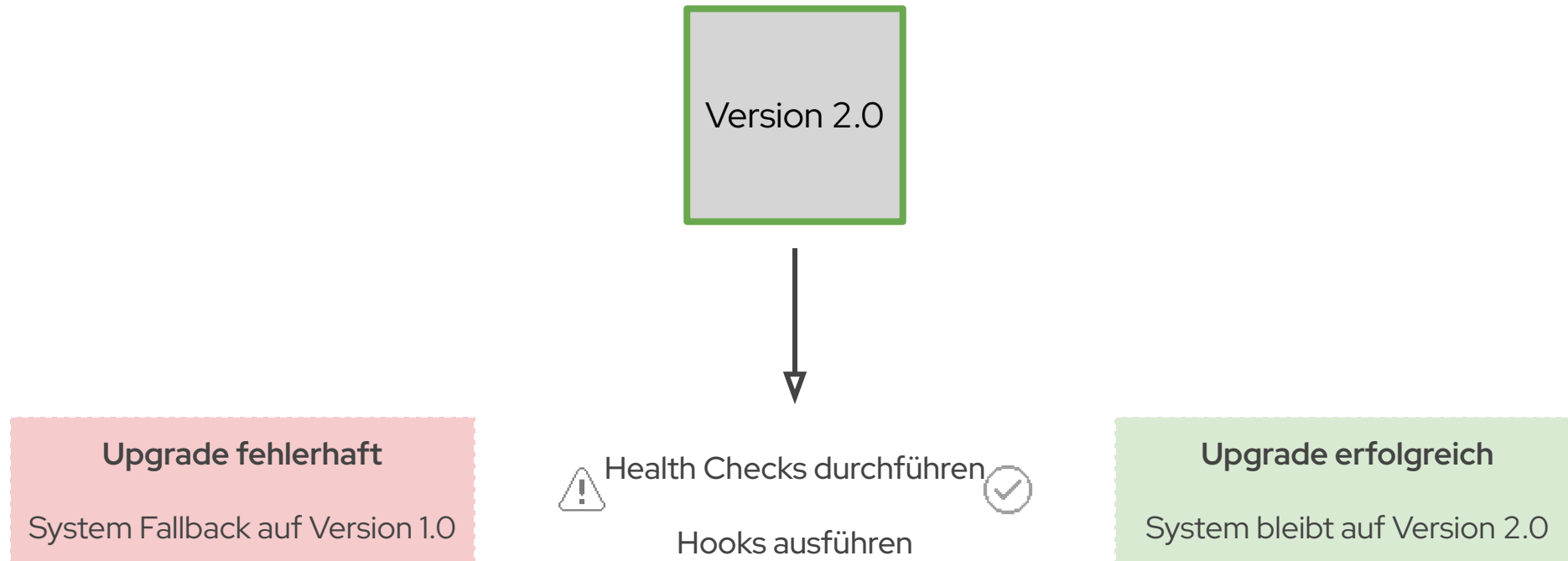
## Lösung: Atomare Updates und intelligenter Rollback

Updates erfolgen entweder ganz oder gar nicht



## Lösung: Intelligenter Rollback

Health Checks rollend fehlgeschlagene Upgrades automatisch zurück

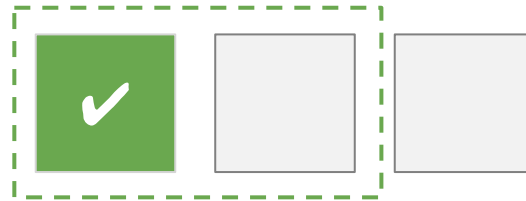


## Lösung: Update und Rollout-Richtlinien

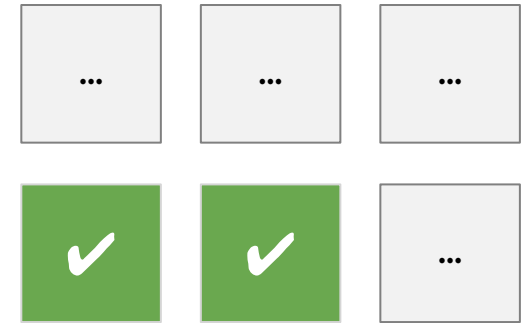
Rollout in kontrollierten, automatisch validierten Wellen



Fleet



Update eines Subsets



Fortsetzen bei Erfolg



## Lösung: Fehlerbudgets

Limit concurrent updates to ensure availability.



Automatischer Abbruch des Rollouts, sobald 20% der Geräte offline sind





**DEMO**

# Herausforderungen mit Edge-App-Deployments



## Ende-zu-Ende Sicherheit

Von der Erstinstallation über den Aufbau an einer Edge-Location bis hin zu Aktualisierungen.



## Konfigurationsabweichungen

Es muss sichergestellt werden, dass sämtliche Endgeräte dieselbe Konfiguration aufweisen - dauerhaft.

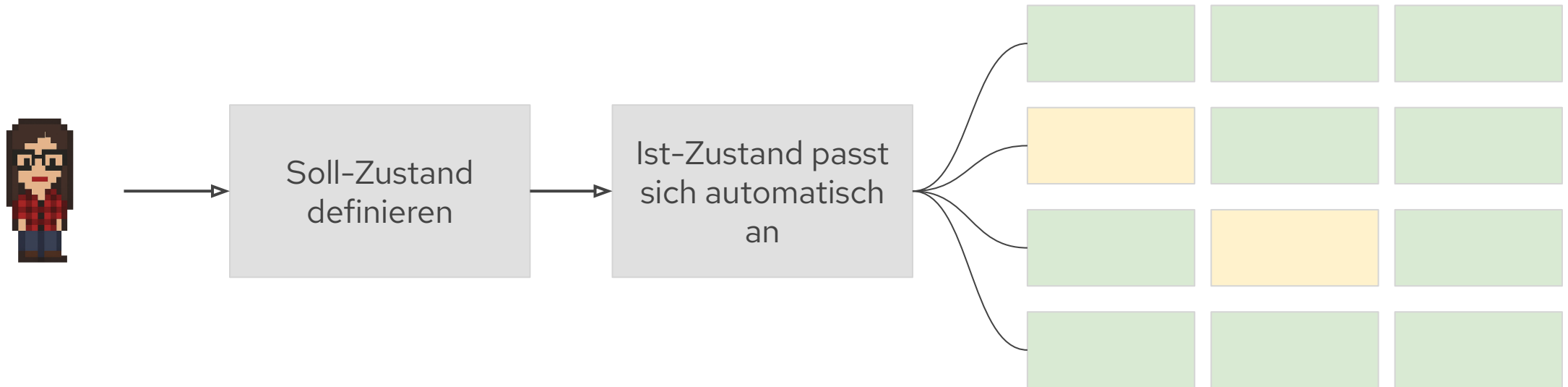


## Troubleshooting

Kein Personal vor Ort oder Personal mit wenig IT-Erfahrung oder adäquaten Werkzeugen.



# Red Hat Edge Manager – Ein Workflow, skalierbar auf Tausende Geräte



## Kommt in die Community. Testet die Tech Preview.

### Dokumentation

Für mehr Informationen steht die [Ansible](#) bzw. [OpenShift](#) Dokumentation bereit.

### Demos

Hands-On Demonstrationen und Workshops stehen auf der Red Hat [Demo Platform](#) bereit.

### Fragen?

Sprechen Sie uns gerne nach der Präsentation an oder kontaktieren Sie uns via E-Mail.





# Jetzt Session bewerten!

Einfach QR-Code scannen,  
Session aus der Liste wählen  
und bewerten. **Vielen Dank!**

[red.ht/rhsc-darmstadt-feedback](https://red.ht/rhsc-darmstadt-feedback)