

OpenShift Enterprise PaaS

Architecture Overview

“How it works”

December 2013



OpenShift PaaS on your choice of cloud or infrastructure...



Public / Hybrid / Private / Virtualization / Bare Metal

The Foundation of OpenShift is Red Hat Enterprise Linux

RHEL

RHEL

RHEL

- OpenShift is built on instances of Red Hat Enterprise Linux (RHEL)
- OpenShift can run anywhere RHEL can run

Public / Hybrid / Private / Virtualization / Bare Metal

An OpenShift Broker Manages Multiple OpenShift Nodes



Broker (RHEL)

OpenShift Broker

– Management and Orchestration Engine



Node (RHEL)

OpenShift Nodes

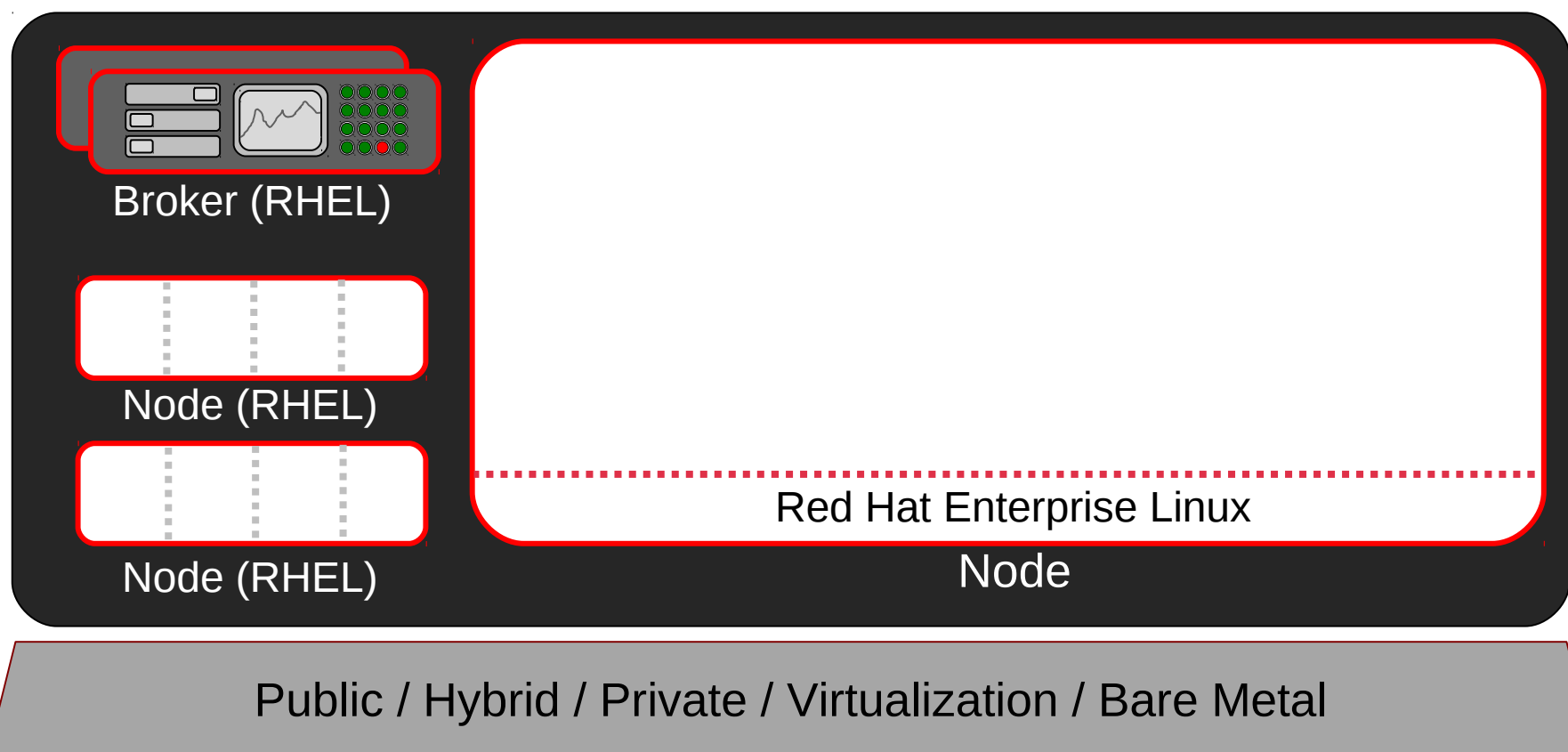
– Application Hosting Infrastructure



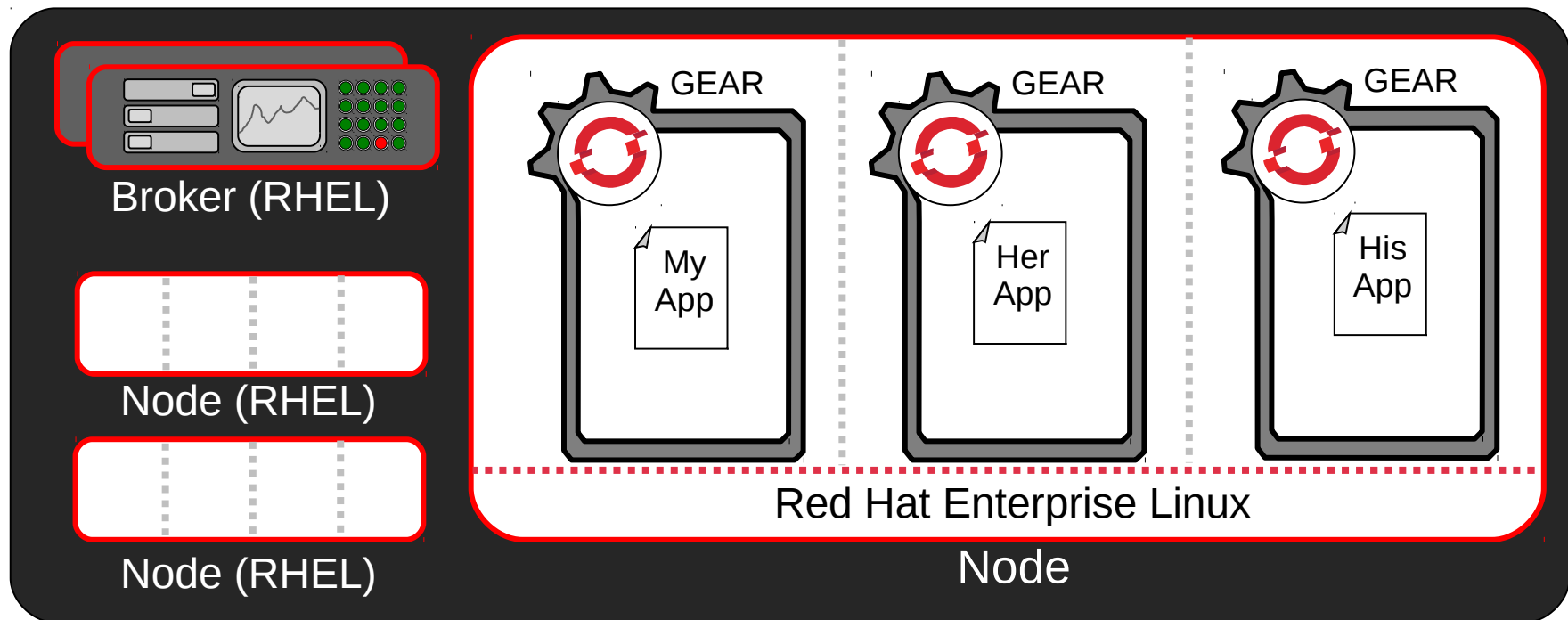
Node (RHEL)

Public / Hybrid / Private / Virtualization / Bare Metal

A Node is an Instance of RHEL

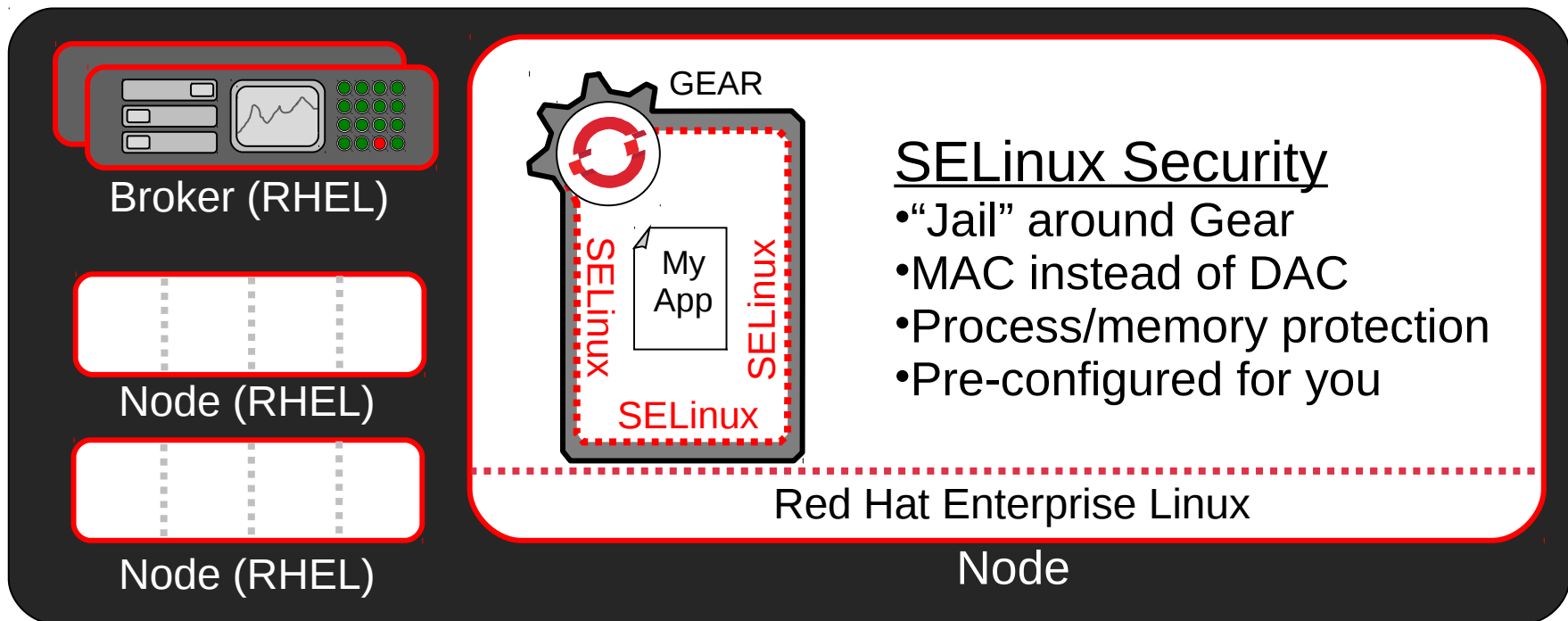


OpenShift User Applications Run in Containers called Gears



Public / Hybrid / Private / Virtualization / Bare Metal

Gears Use SELinux for Pre-Configured NSA-Grade Security

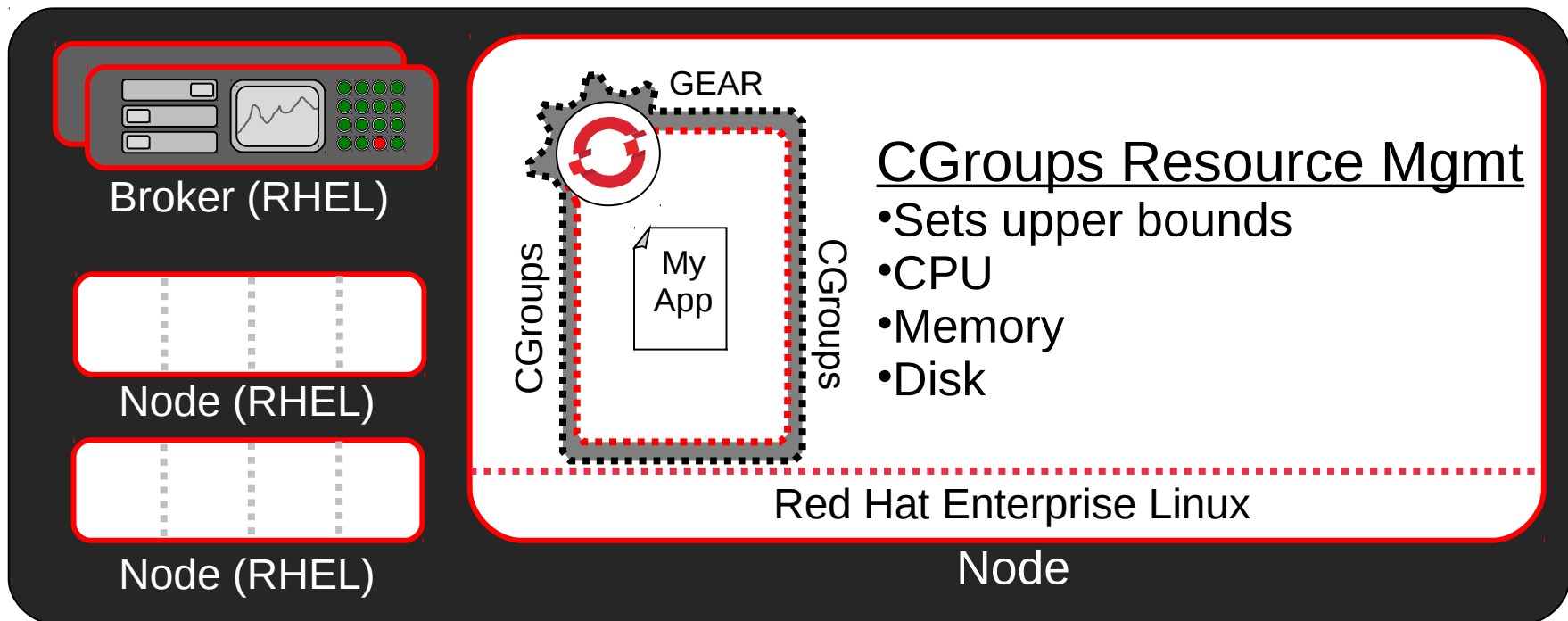


SELinux Security

- “Jail” around Gear
- MAC instead of DAC
- Process/memory protection
- Pre-configured for you

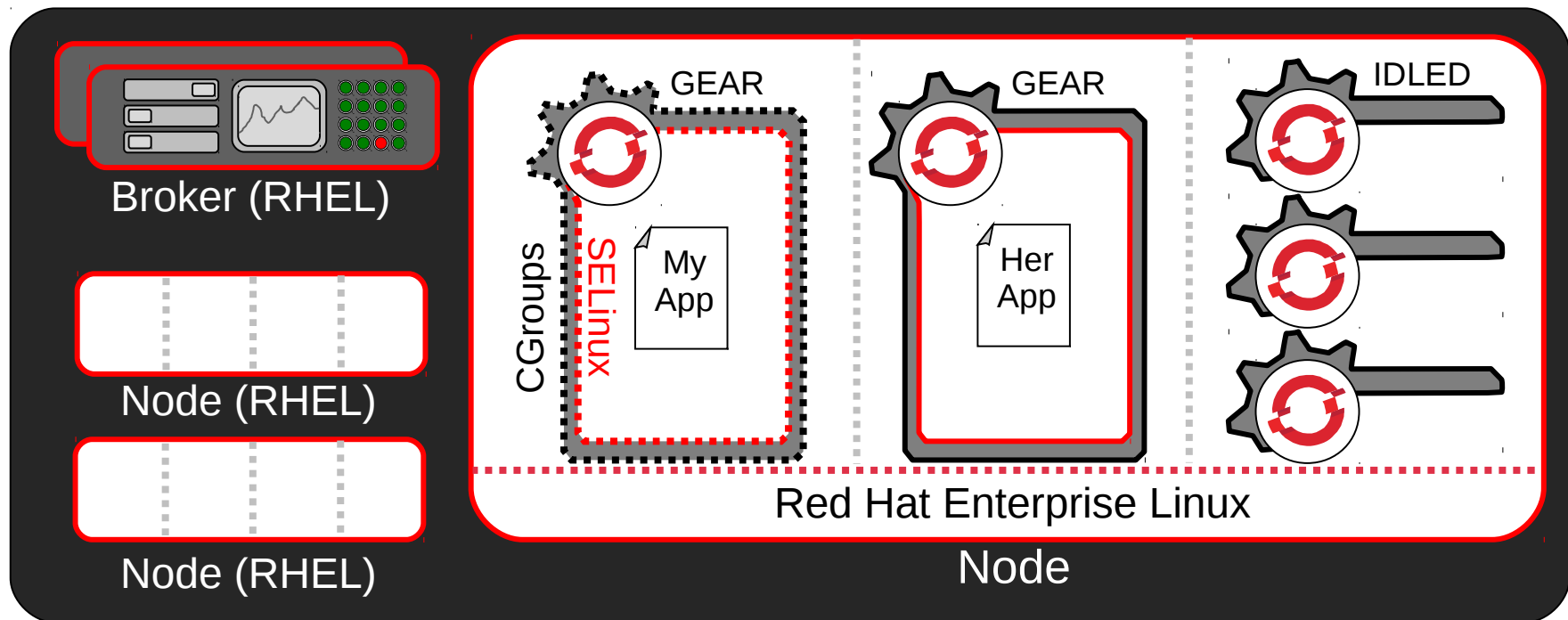
Public / Hybrid / Private / Virtualization / Bare Metal

Gears Use Linux CGroups for Resource Management



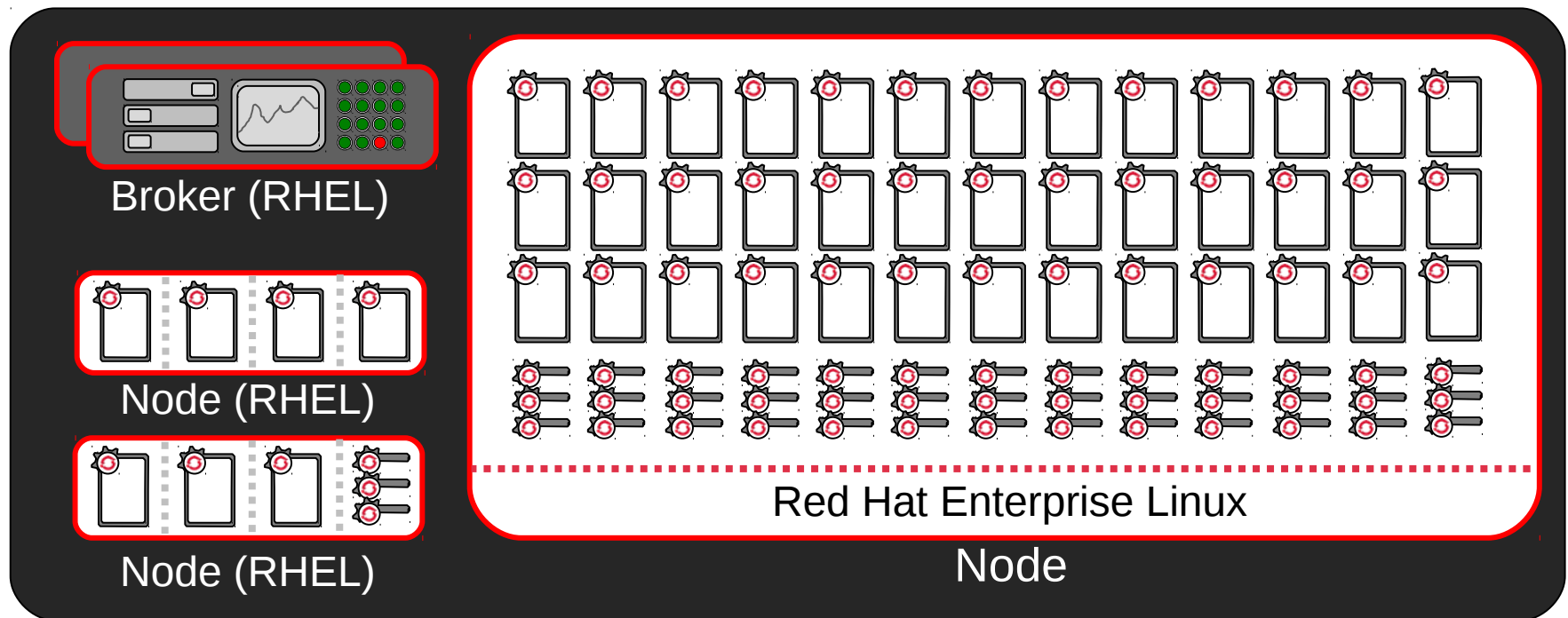
Public / Hybrid / Private / Virtualization / Bare Metal

Idle Gears Can Be “de-hydrated” by the OpenShift Broker



Public / Hybrid / Private / Virtualization / Bare Metal

OpenShift Multi-tenancy Provides Density, Efficiency, and Security

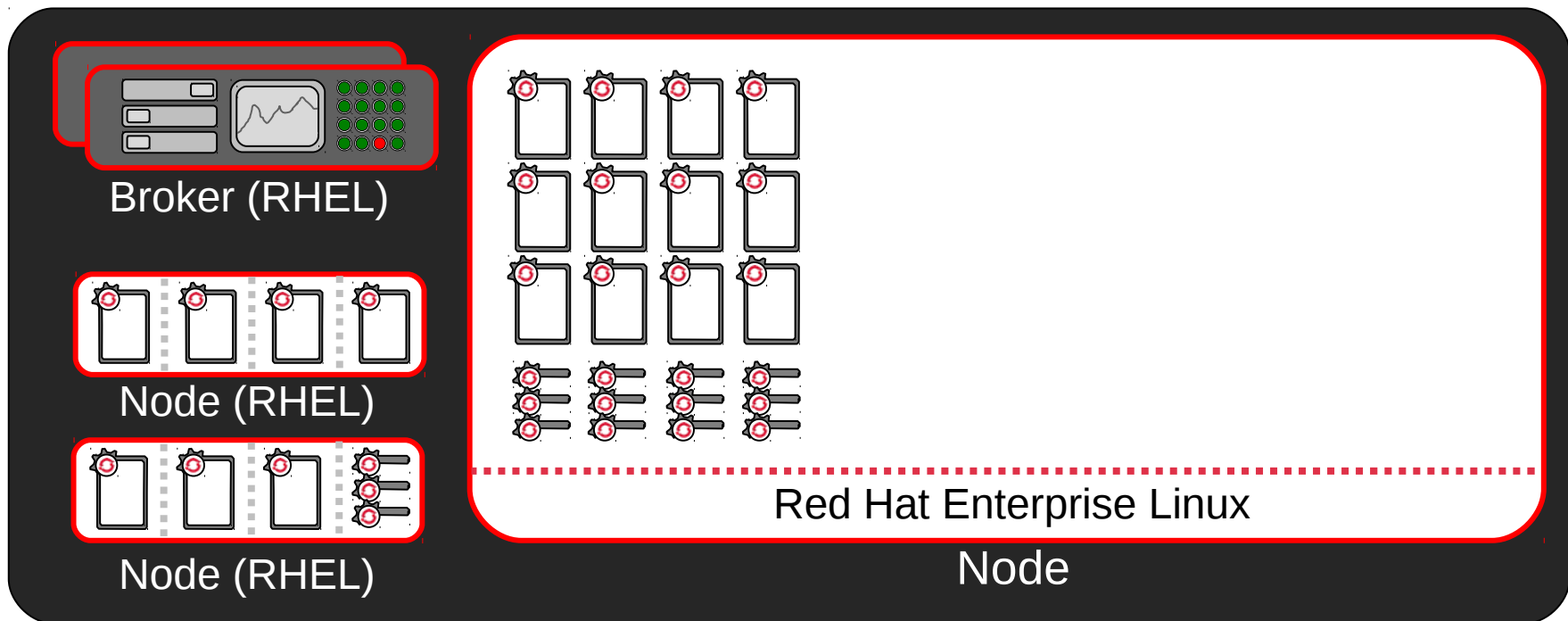


Public / Hybrid / Private / Virtualization / Bare Metal

Developer Workflow



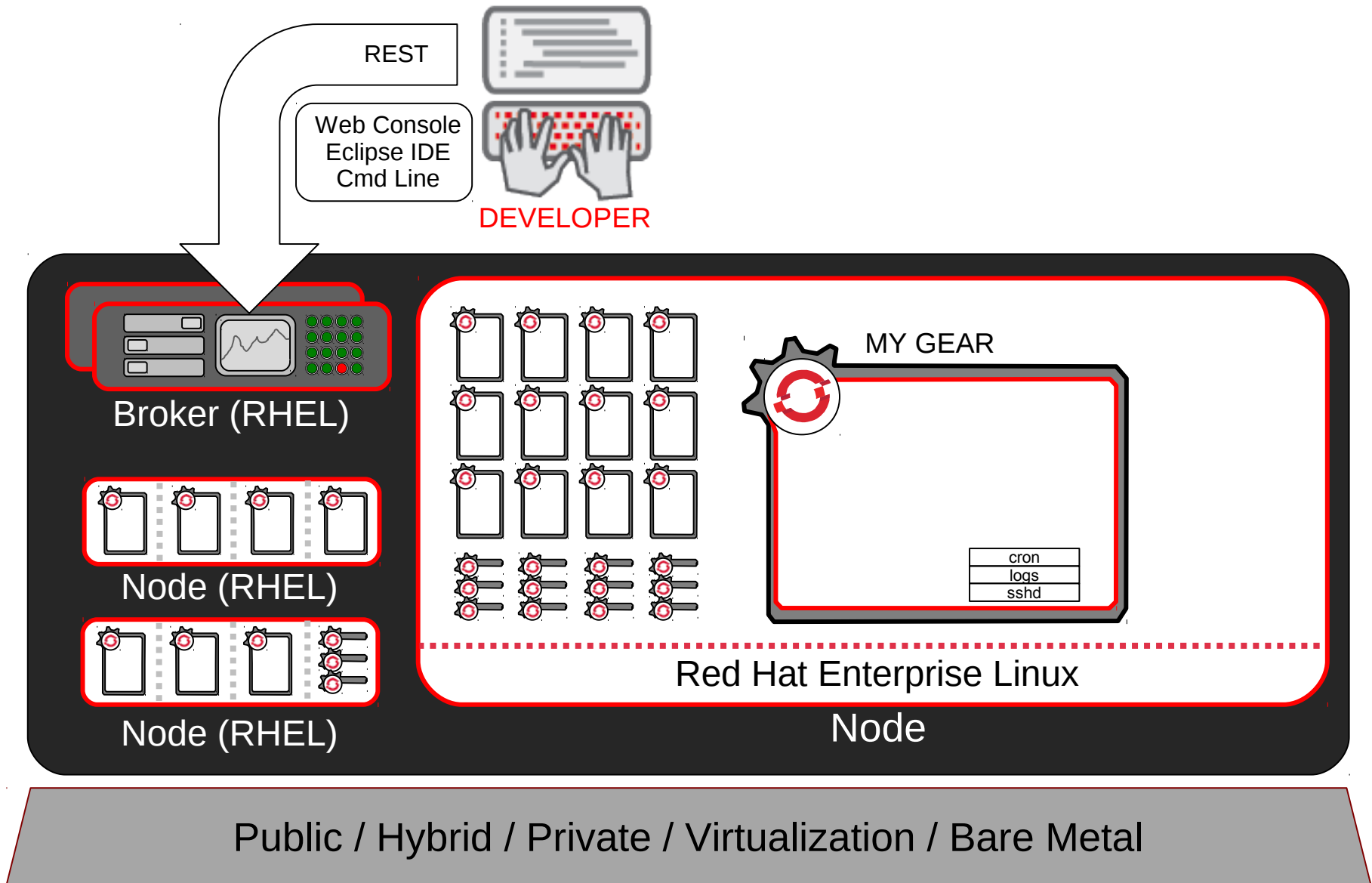
A Developer creates a new application →
OpenShift creates a GEAR



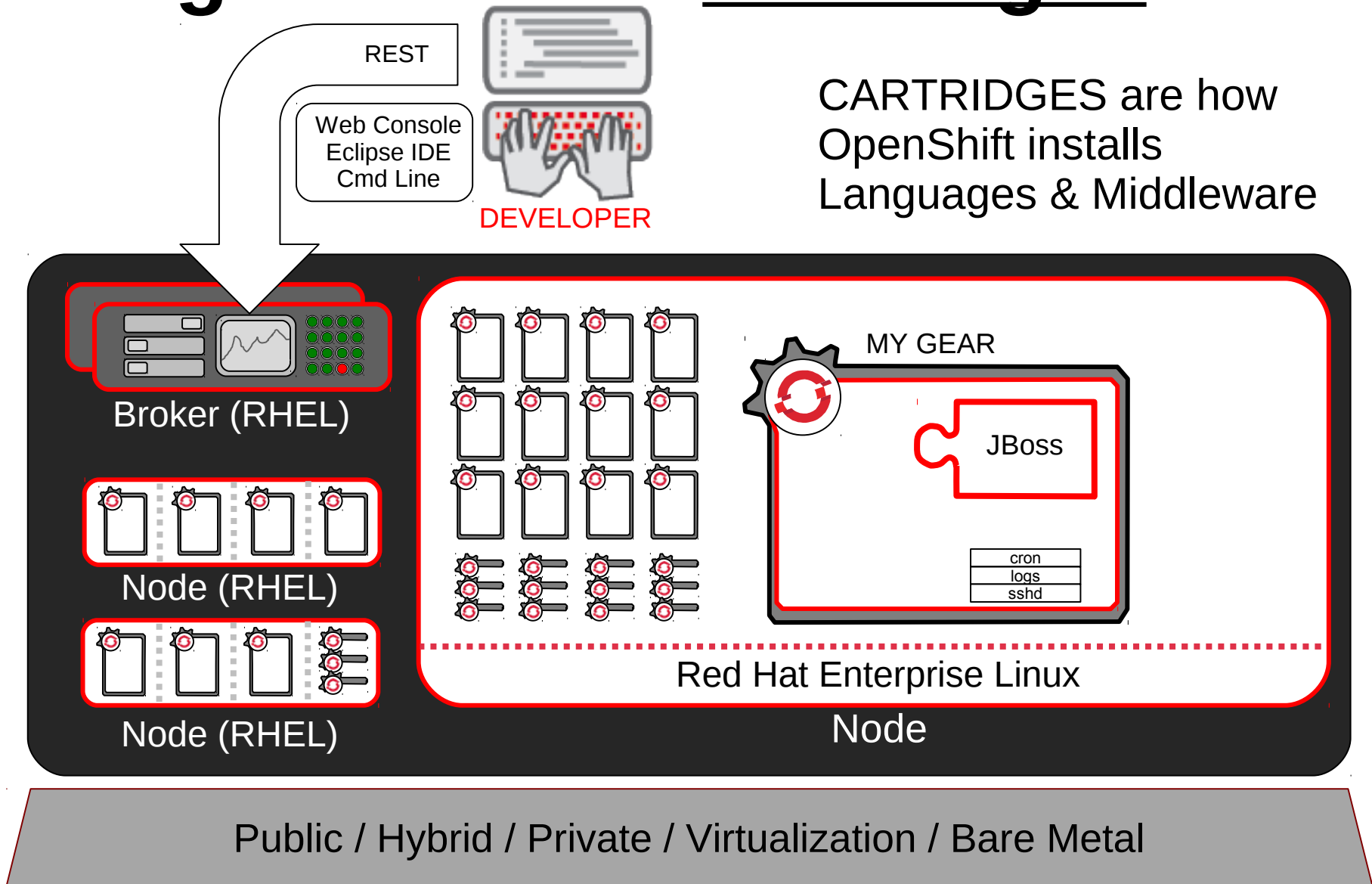
Public / Hybrid / Private / Virtualization / Bare Metal

Gears Created via Web, CLI, Eclipse

-RESTful call to Broker

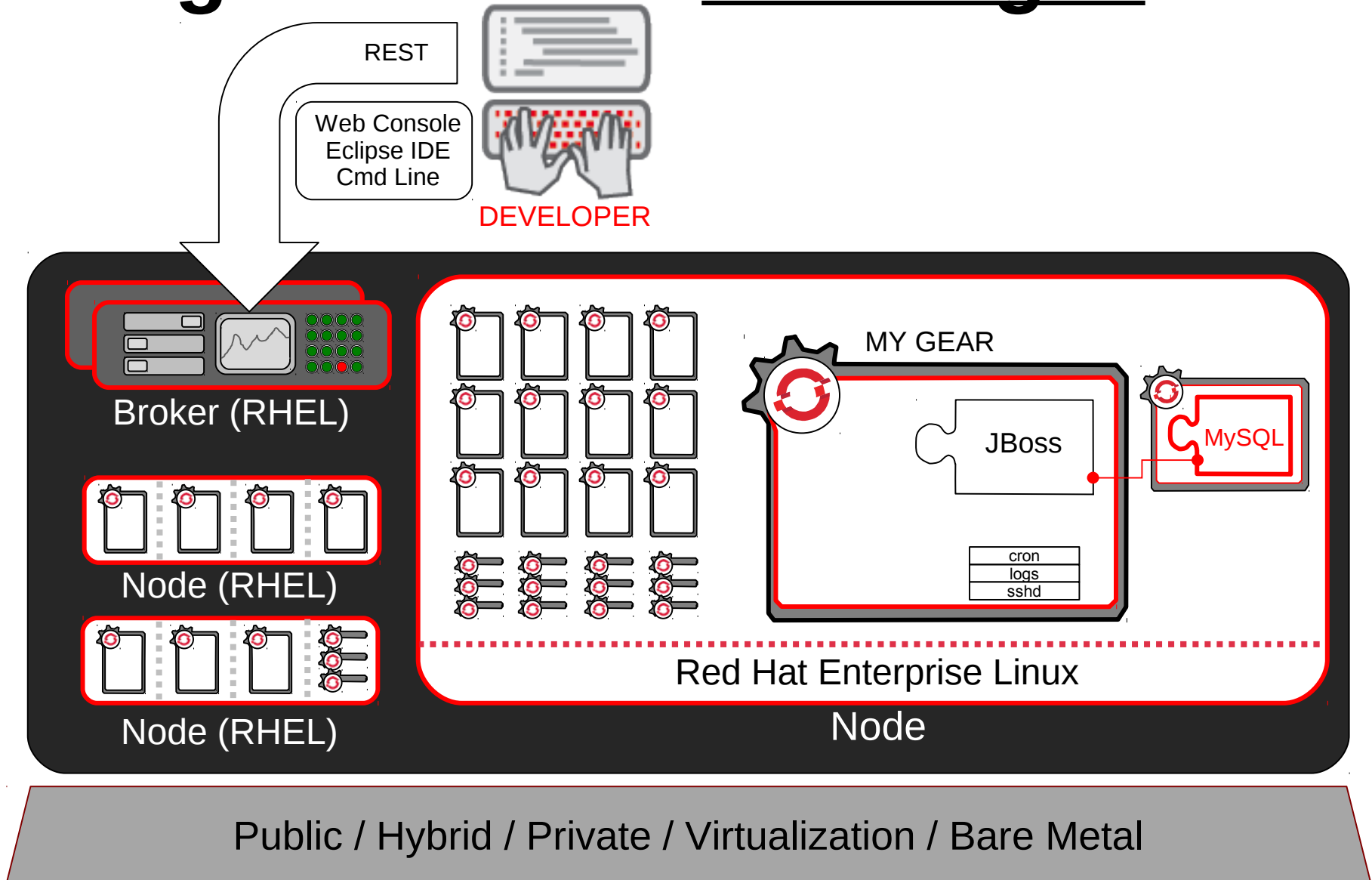


OpenShift Automates Gear Configuration via Cartridges

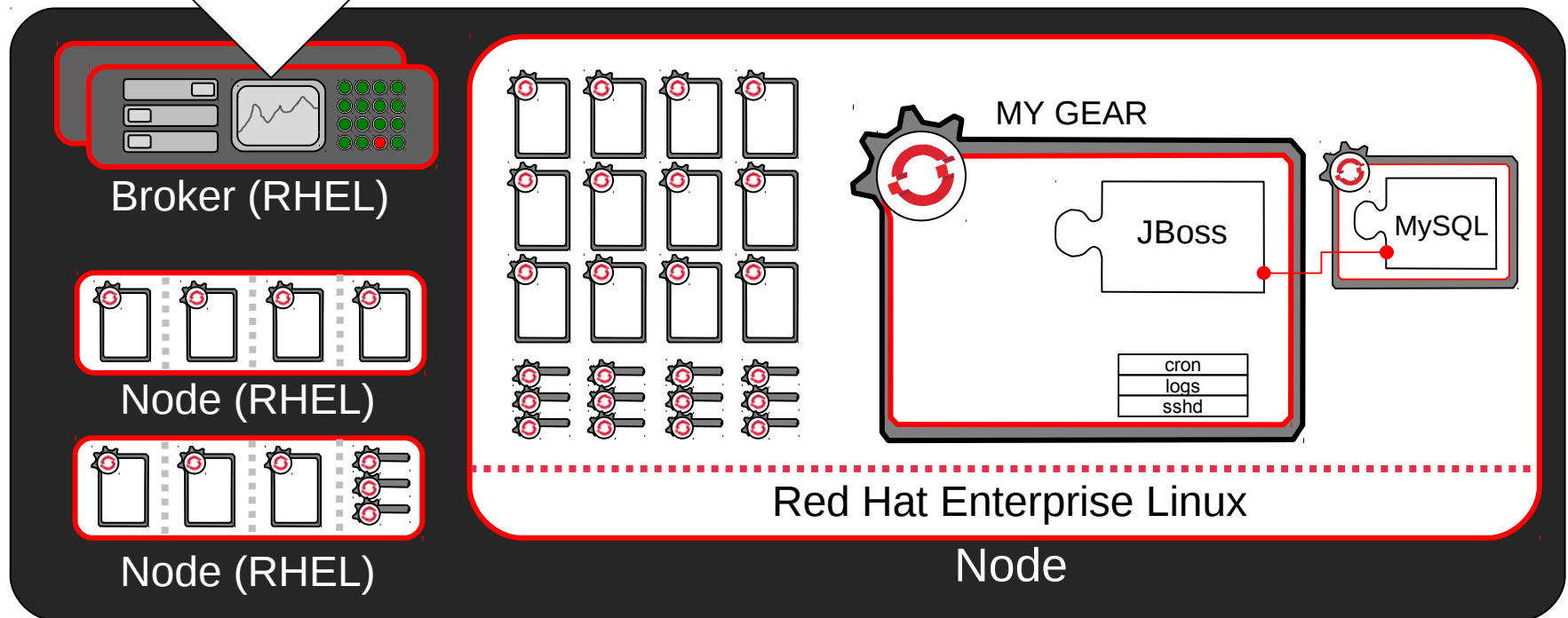
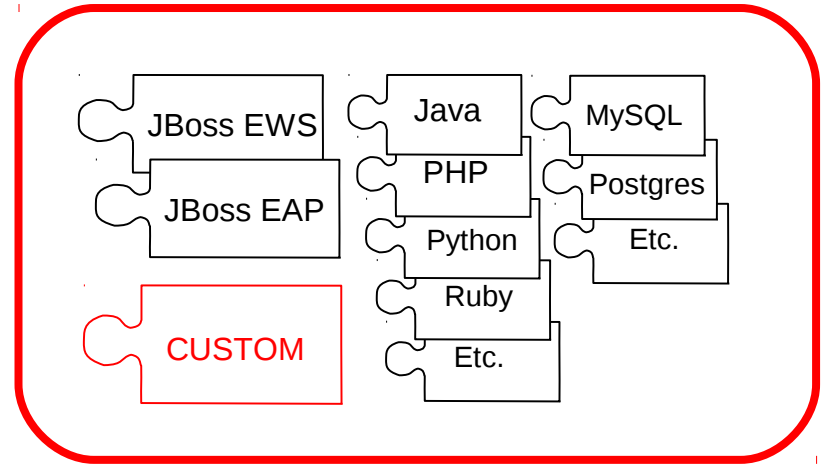
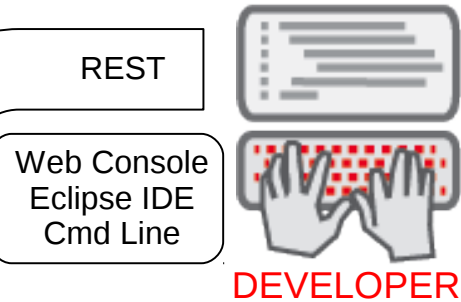


CARTRIDGES are how OpenShift installs Languages & Middleware

OpenShift Automates Gear Configuration via Cartridges



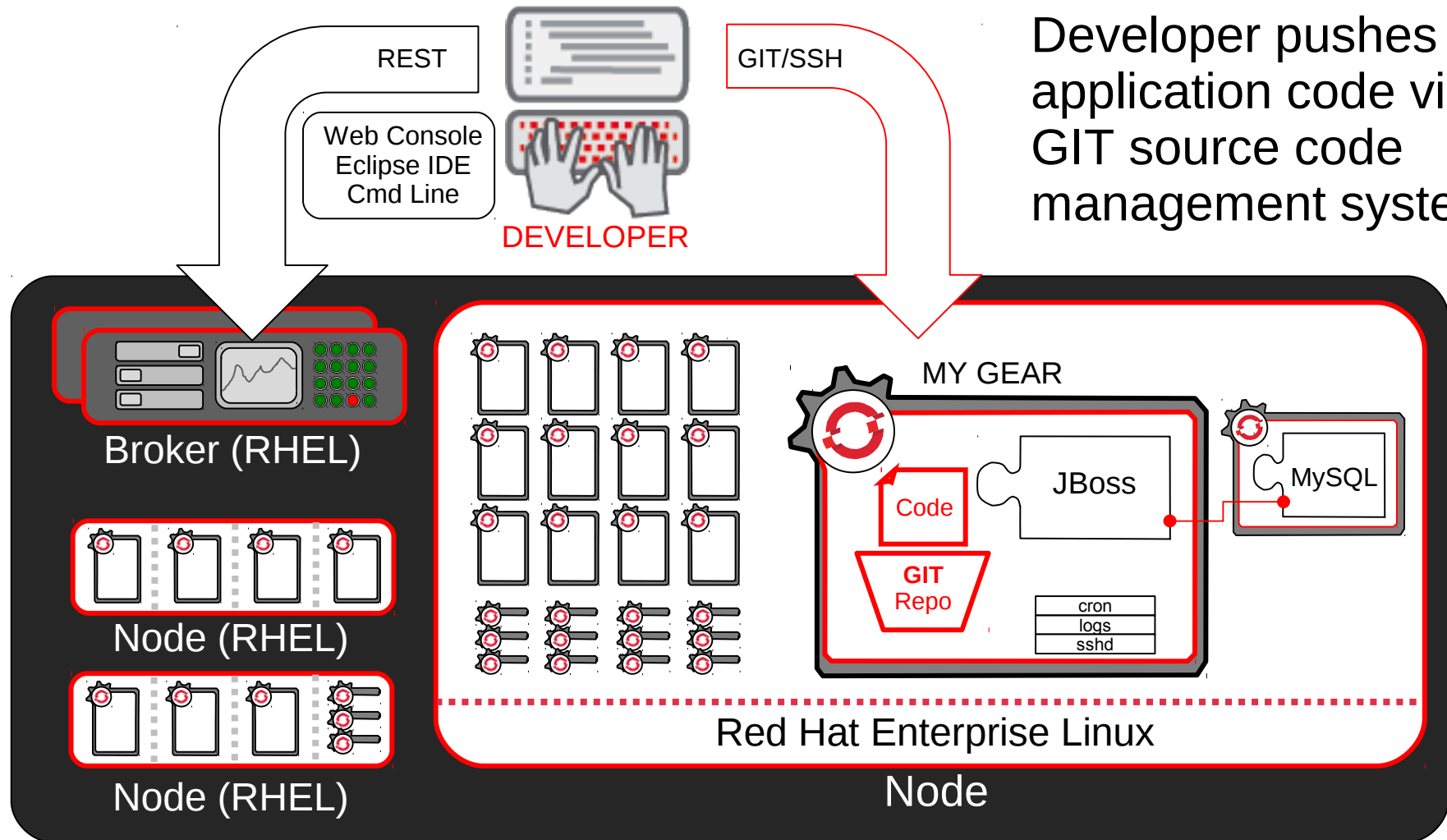
Cartridge Types



Public / Hybrid / Private / Virtualization / Bare Metal

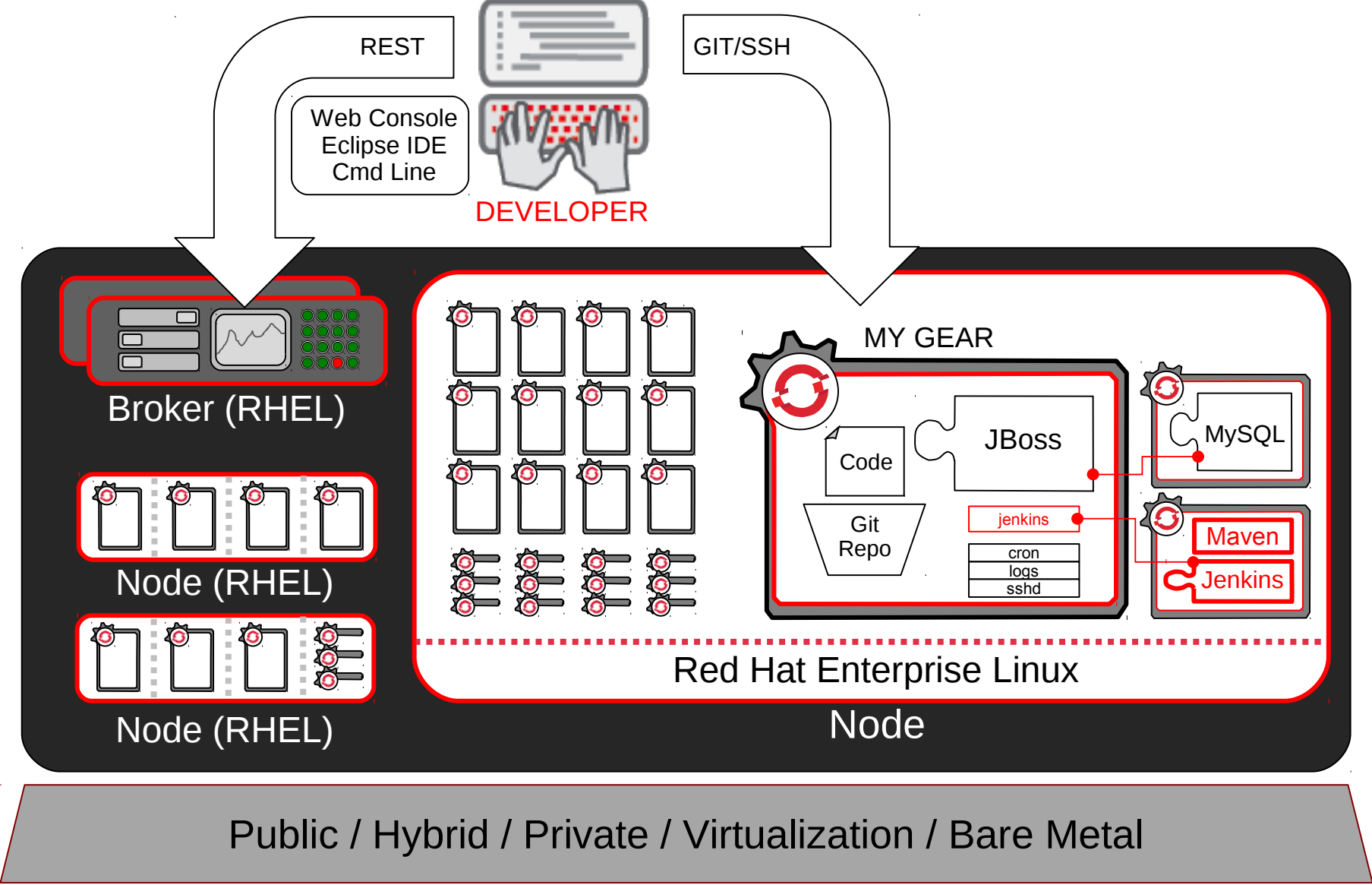
Now, Code and Push

Developer pushes application code via GIT source code management system

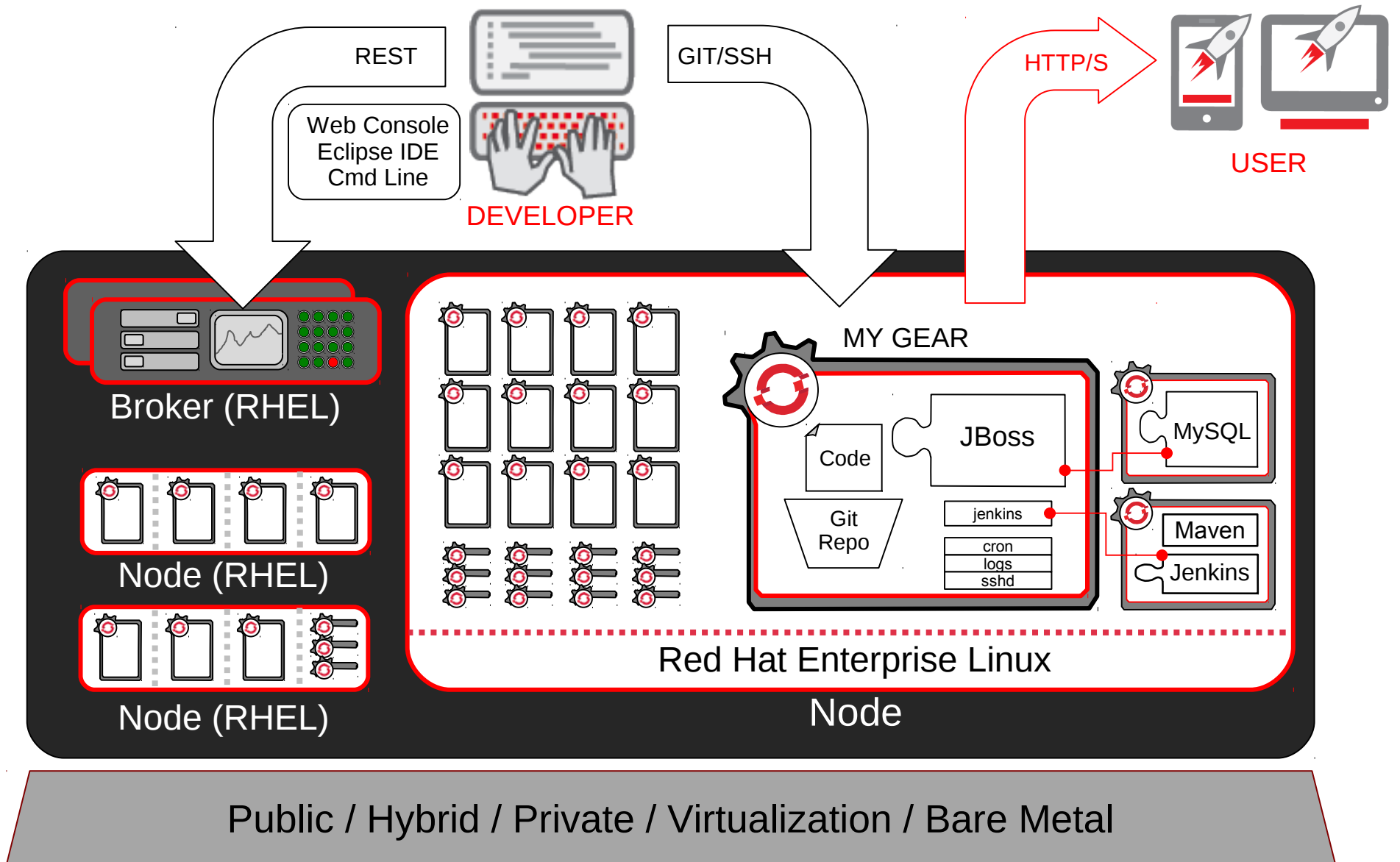


Public / Hybrid / Private / Virtualization / Bare Metal

OpenShift Automates Build, Test with Maven and Jenkins for CI

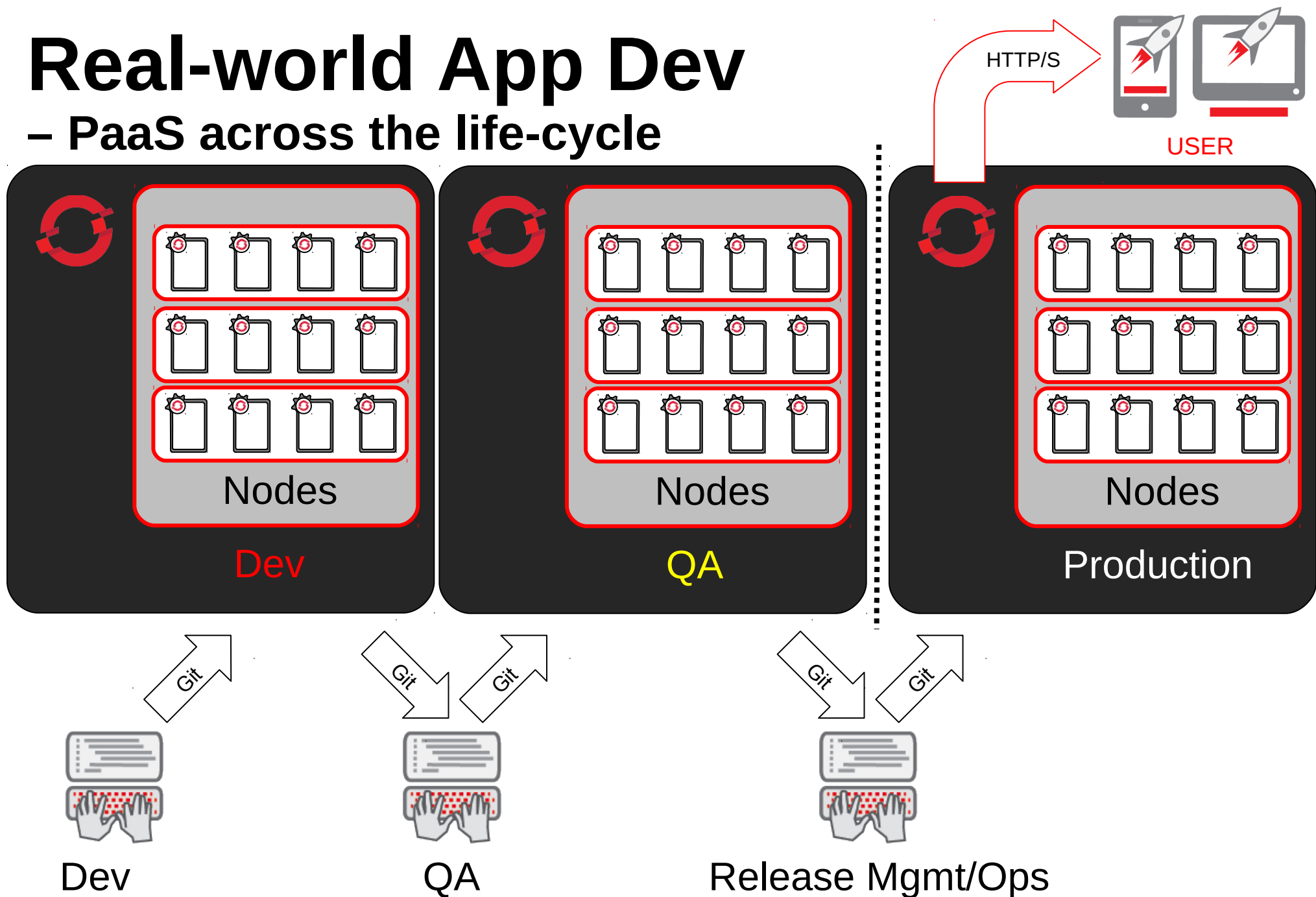


HTTP(s) Served from Gears

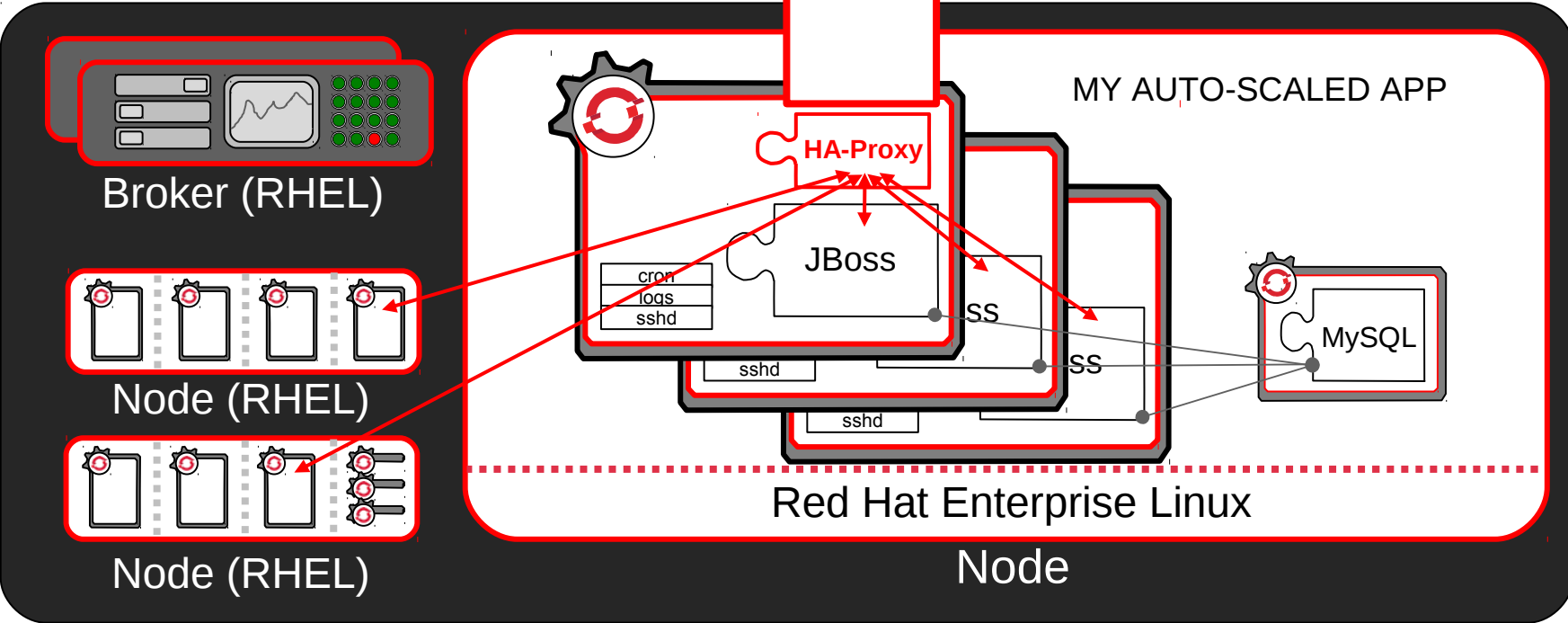


Real-world App Dev

- PaaS across the life-cycle



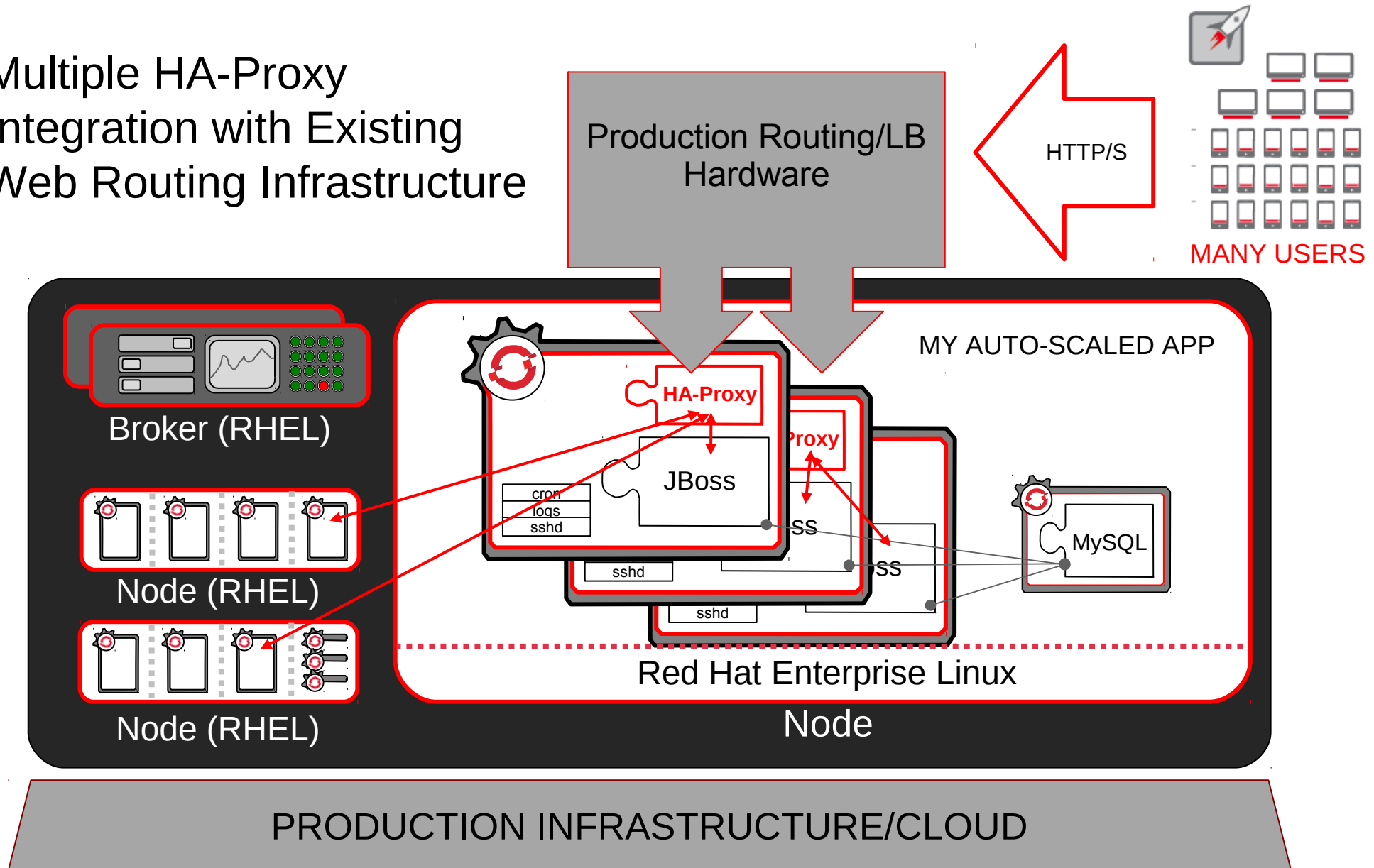
OpenShift Automates Application Scaling!



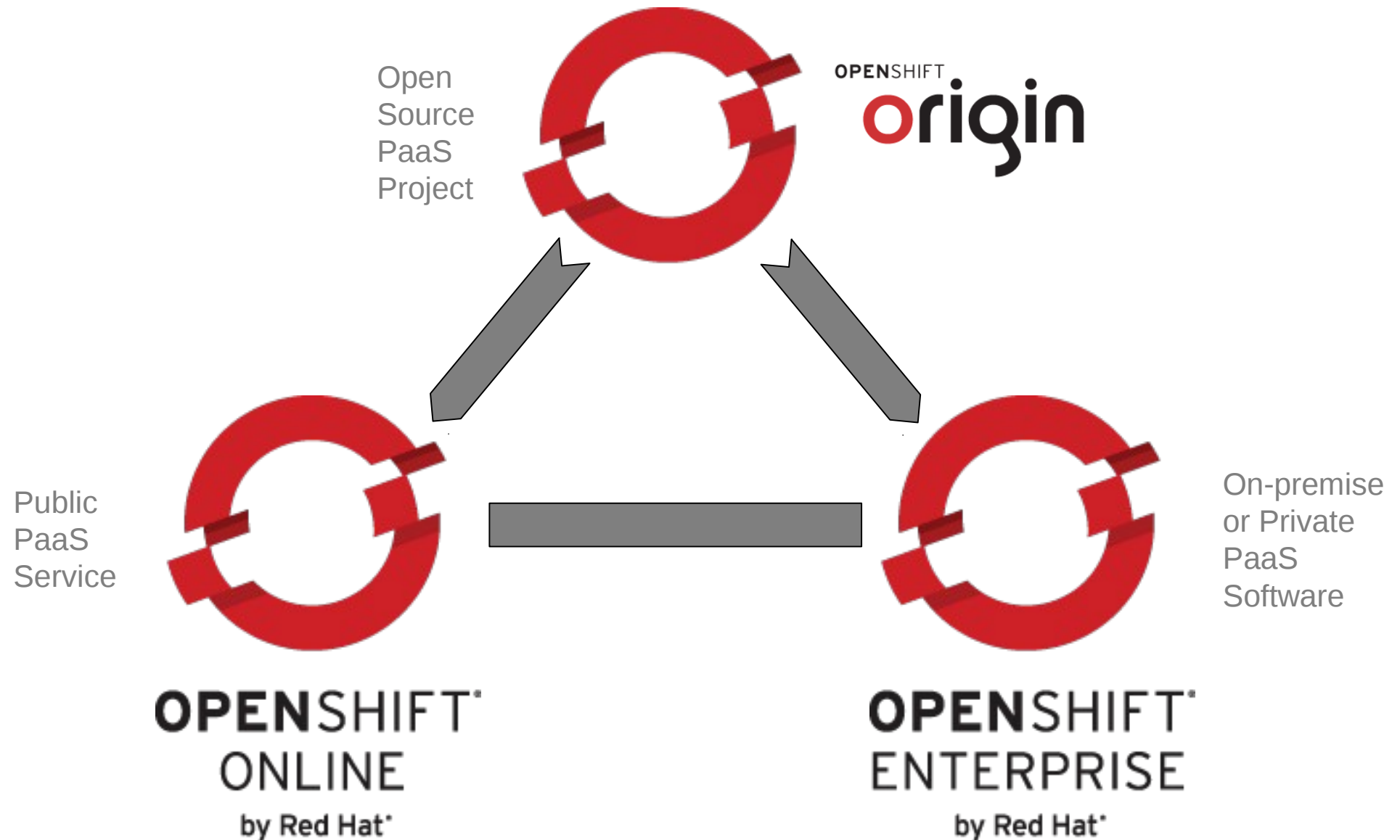
PRODUCTION INFRASTRUCTURE/CLOUD

External Load Balancer Integration

- Multiple HA-Proxy
- Integration with Existing Web Routing Infrastructure



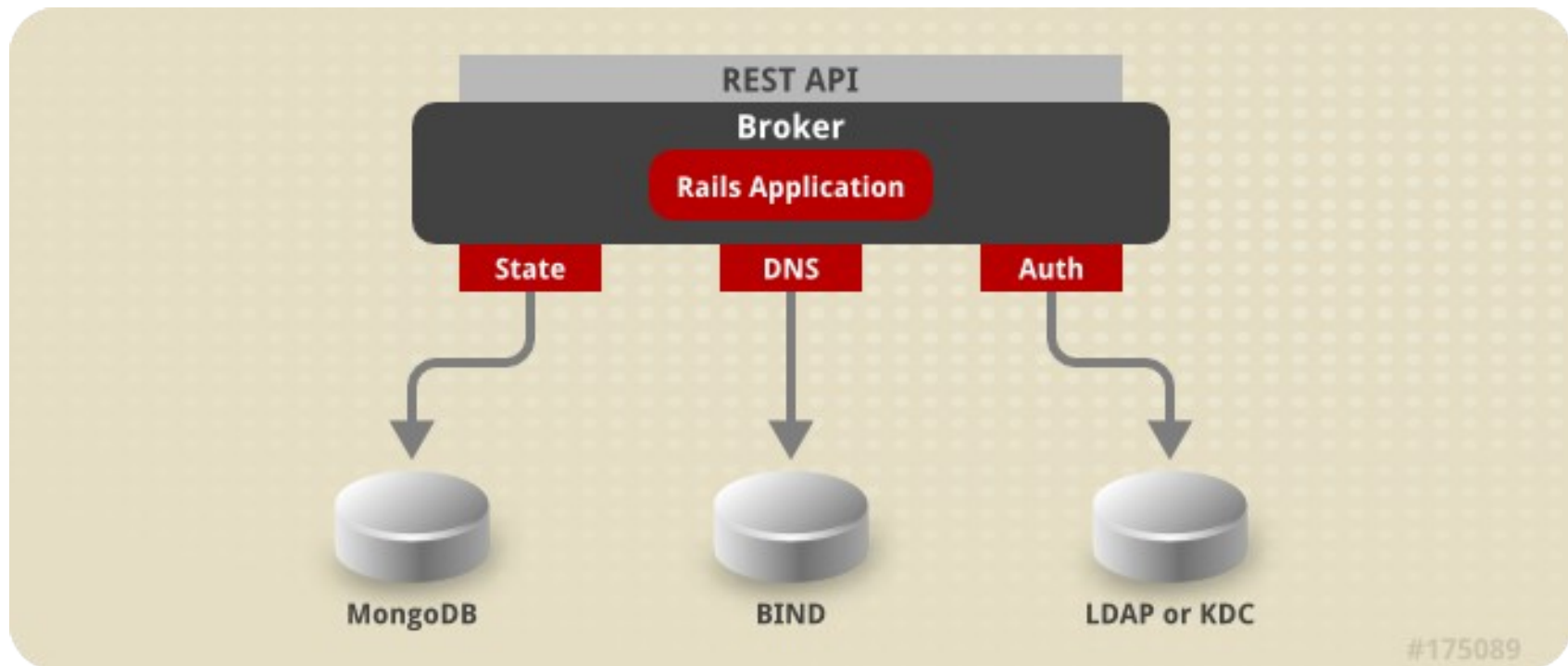
Red Hat's PaaS Strategy



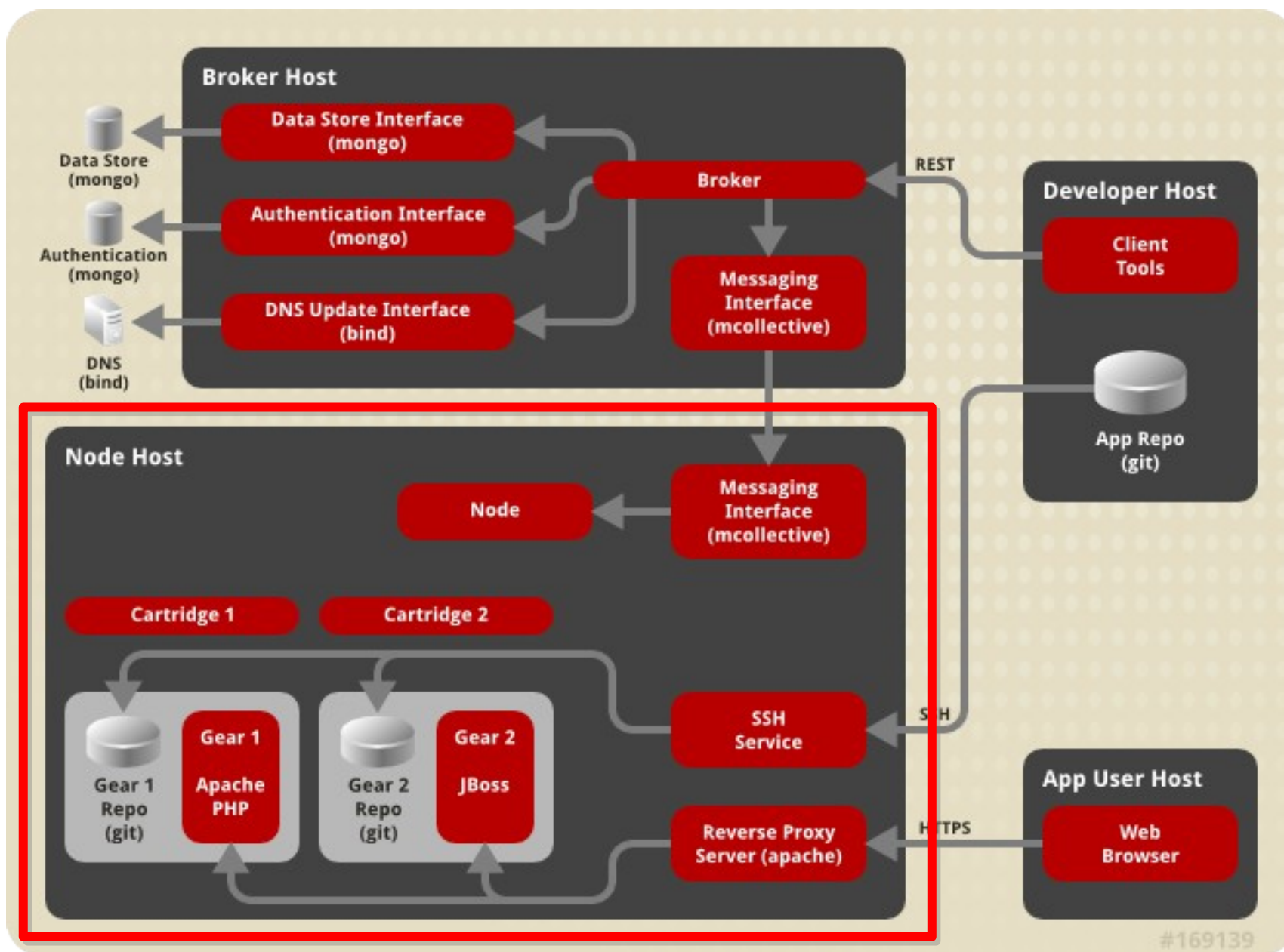
The OpenShift Broker

- The PaaS Controller Subsystem

The Broker is responsible for state, DNS, and authentication.



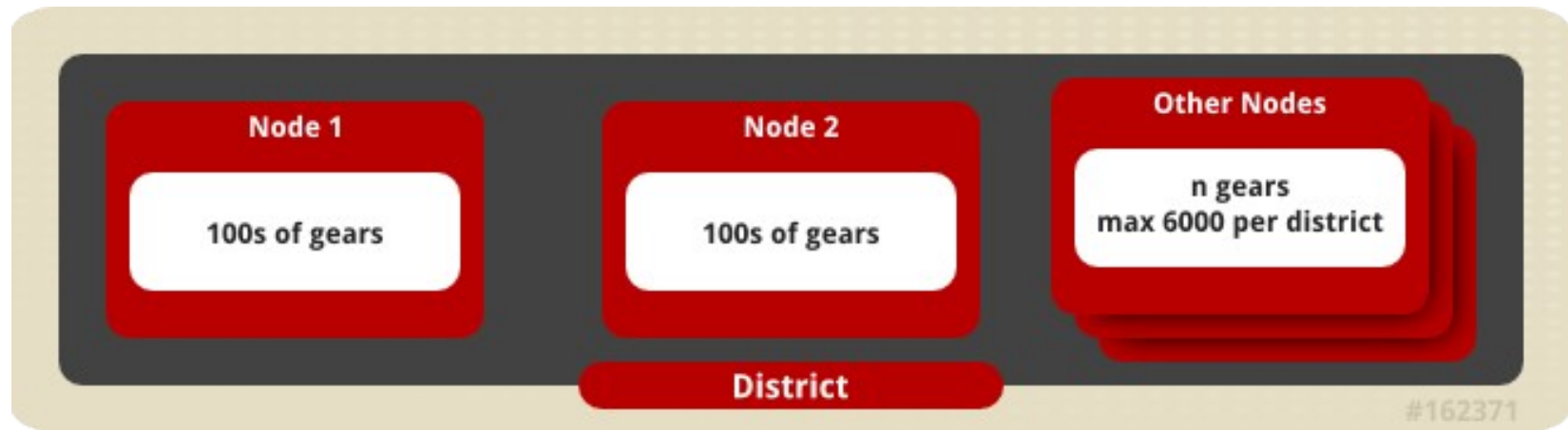
OpenShift Architecture



The OpenShift Nodes

– The Application Host Systems

The node hosts are responsible for holding application gears.

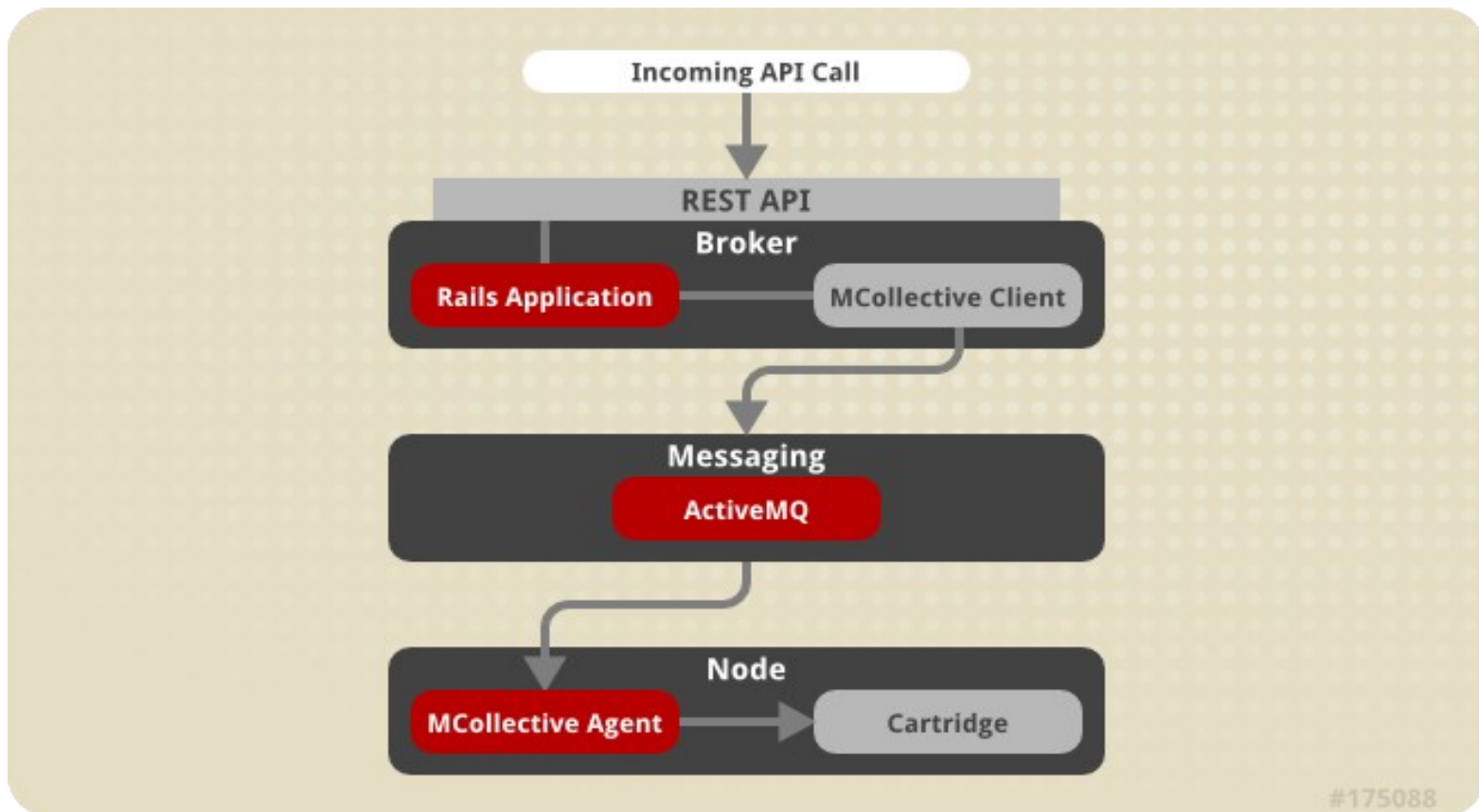


Control Communications

– How Commands Are Delivered To the PaaS

Communication from external clients occurs through the REST API

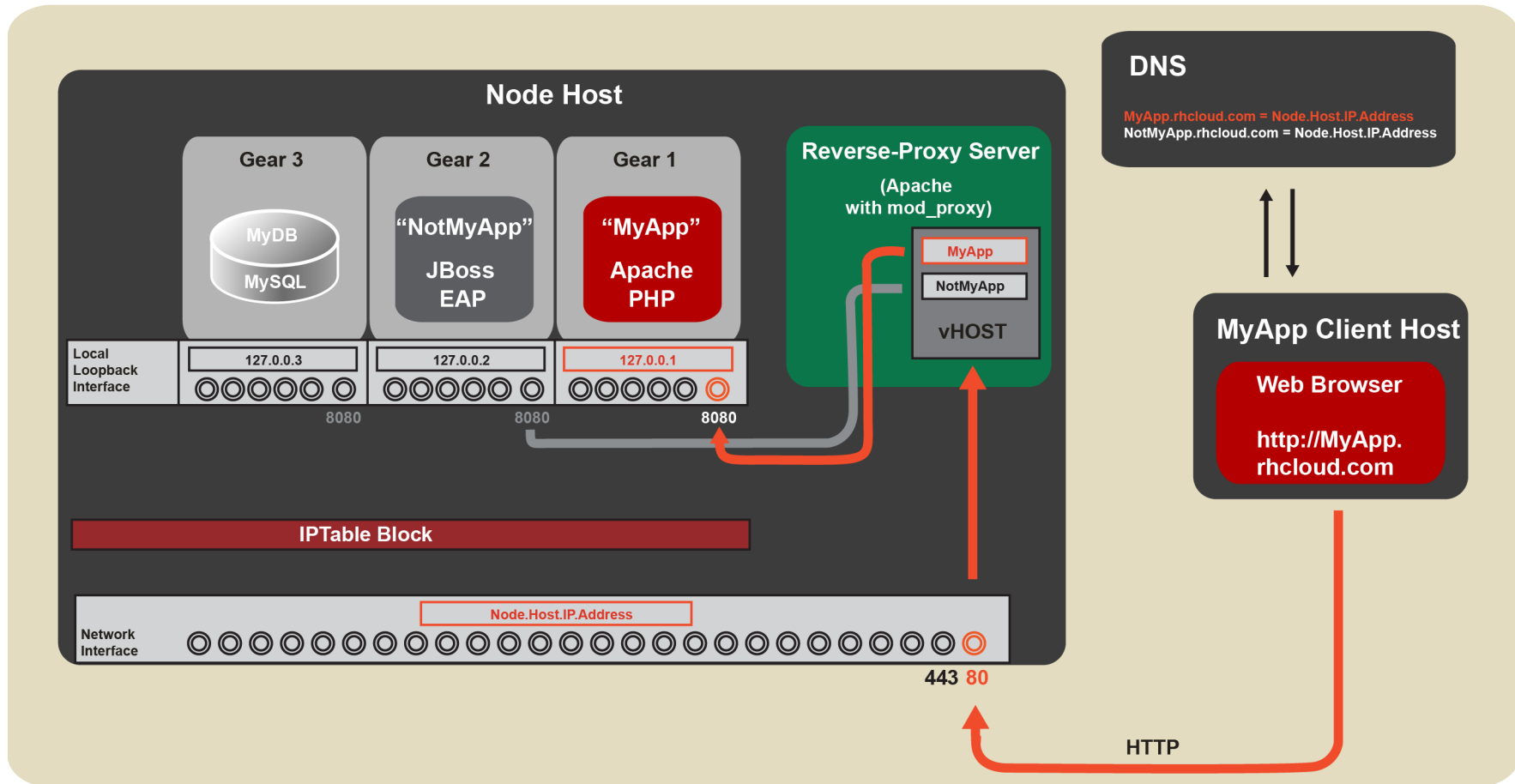
The Broker then communicates through the messaging service to nodes



Application Communications – Part 1

– Incoming Requests to User Applications

OpenShift Enterprise Networking - Part 1: Incoming Application Traffic



Application Communications – Part 2

– Inter-Gear Communications

OpenShift Enterprise Networking - Part 2: Inter-Gear TCP Communication

