

HOW OPEN SOURCE IS DRIVING DEVOPS INNOVATION

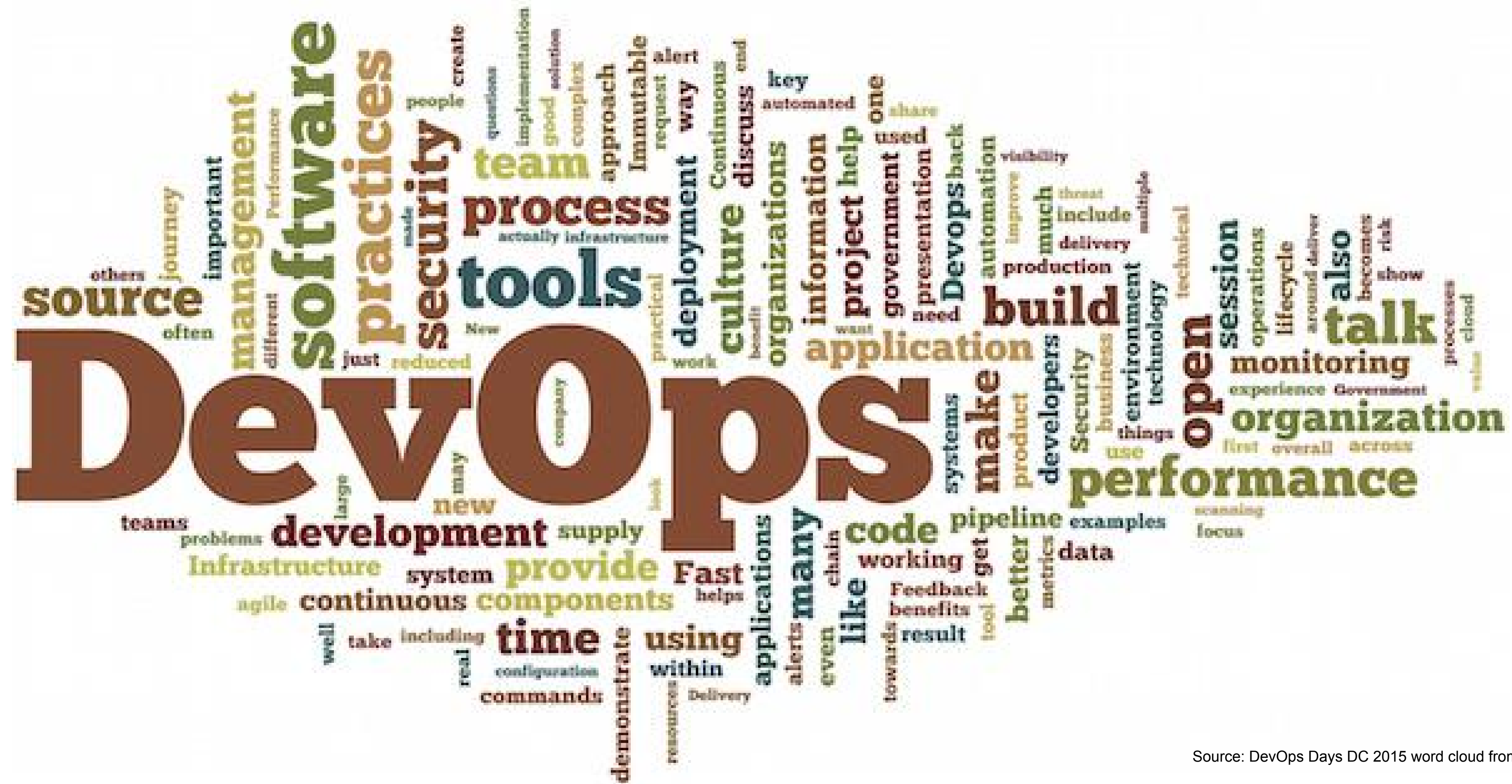
Gordon Haff @ghaff

William Henry @ipbabble

Cloud & DevOps Product Strategy, Red Hat

17 August 2015

What is DevOps?



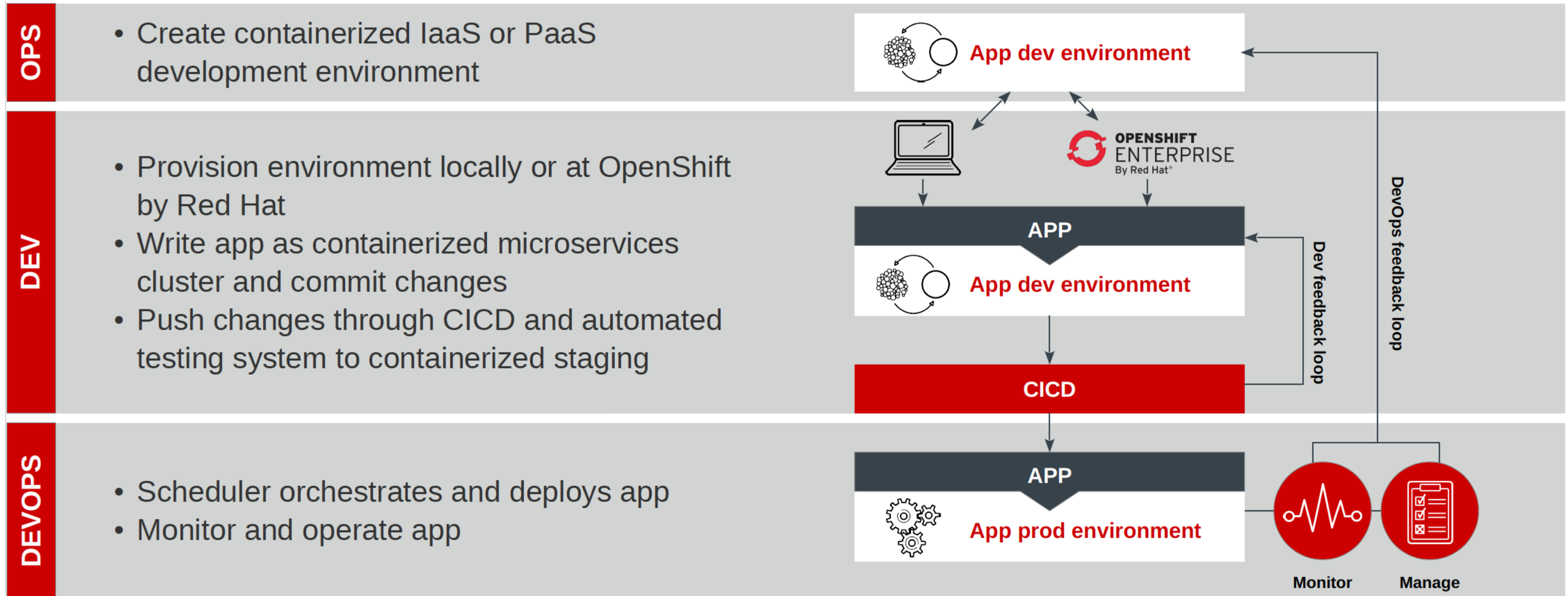
Source: DevOps Days DC 2015 word cloud from Open Spaces.

**DevOps applies open source
principles and practices with...**

TOOLS drawing from innovative development communities



A typical DevOps workflow



Tools for operations*/infrastructure

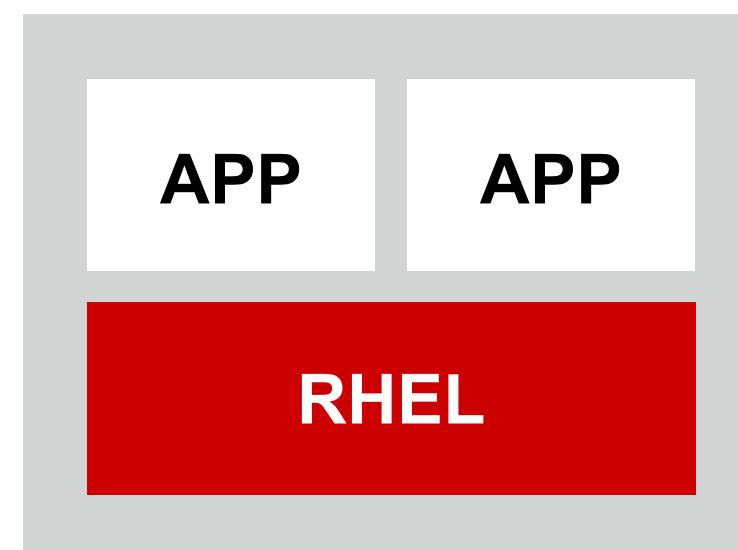
Containerized infrastructure
Orchestration
Policy-based governance
Configuration management
Automation
Packaging
Updates

* But they bleed into developer land!



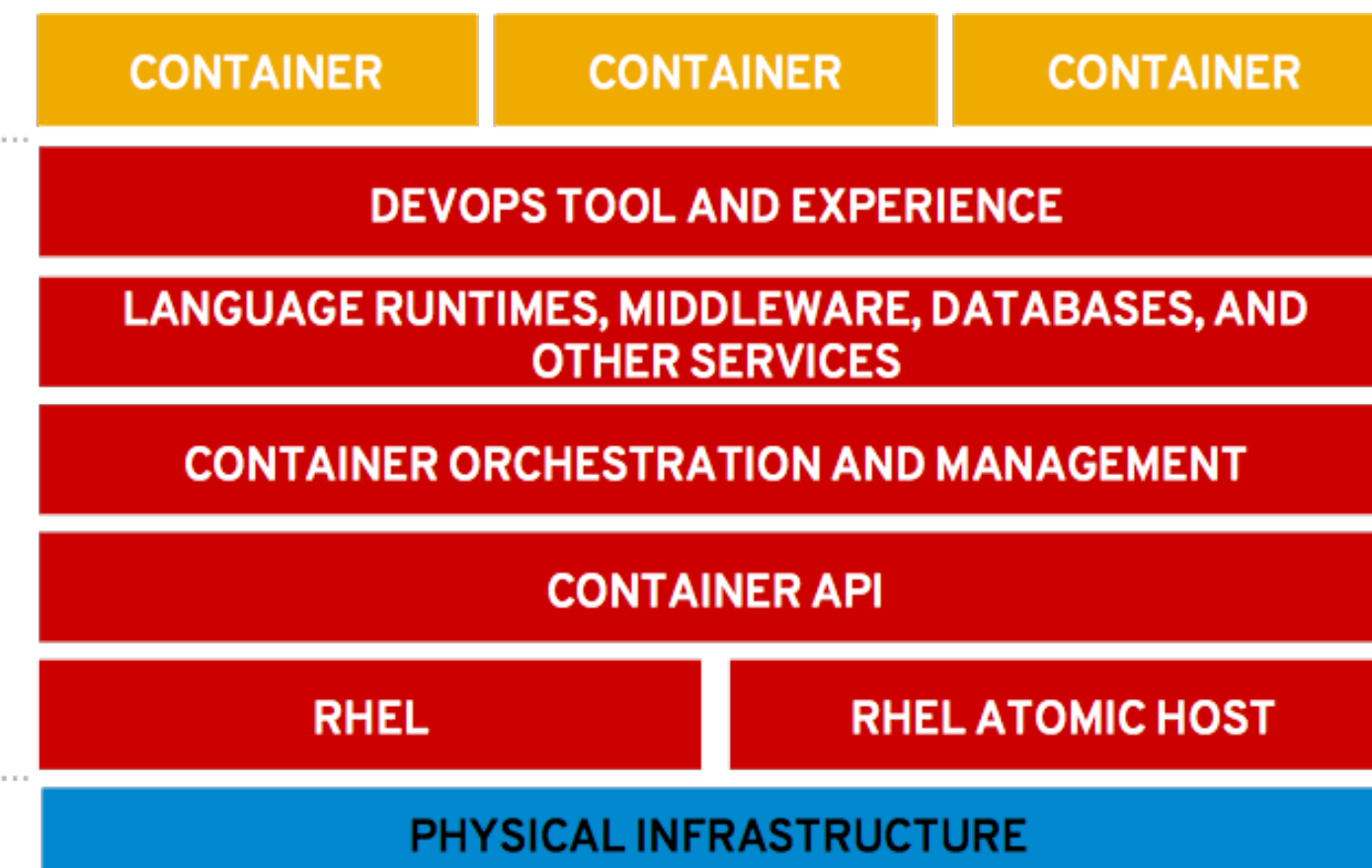
Infrastructures evolve for cloud-native

- Monolithic app container
- Scale up by adding hardware resources
- Limited scale out through clustering



SINGLE-HOST APPS

- Distributed, networked, containerized services
- Scale out by orchestrating services
- *Faster iteration and release*
- *More robust*



**RED HAT®
ATOMIC ENTERPRISE
PLATFORM**

MULTI-HOST APPS

Infrastructure layer: A Cloud Platform for Microservice Cloud Apps

Provision apps from
service catalog

**OPS MANAGEMENT AND
SERVICE CATALOG**
(e.g. RED HAT CLOUDFORMS)

**CONTENT, ENTITLEMENT,
AND LIFECYCLE**
(e.g. RED HAT SATELLITE)

Orchestrate and place apps

SERVICE SCHEDULER/ORCHESTRATOR
(e.g. KUBERNETES)

Run composed microservices
in containers

Red Hat
CloudForms
Monitoring
Docker
Image

Red Hat
CloudForms
Orchestration
Docker
Image

Red Hat
Satellite
Content
Docker
Image

**RHEL ATOMIC
HOST GUEST**

Red Hat
JBoss
AMQ
Docker
Image

App DB
Docker
Image

Red Hat
JBoss
BRMS
Docker
Image

**RHEL ATOMIC
HOST GUEST**

PaaS
(e.g. OPENSIFT)

Provide dynamic, programmable
infrastructure

RED HAT ENTERPRISE LINUX OPENSTACK PLATFORM
COMPUTE STORAGE NETWORK
(e.g. OPENSTACK, CEPH, OPEN DAYLIGHT)

Tools for developer* workflows

Collaboration

CI/CD

Issue tracking

Source code control

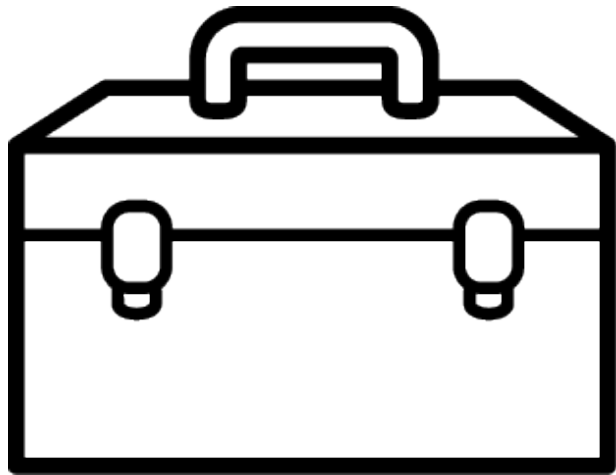
Code review

Platform-as-a-Service

* But they bleed into ops land!



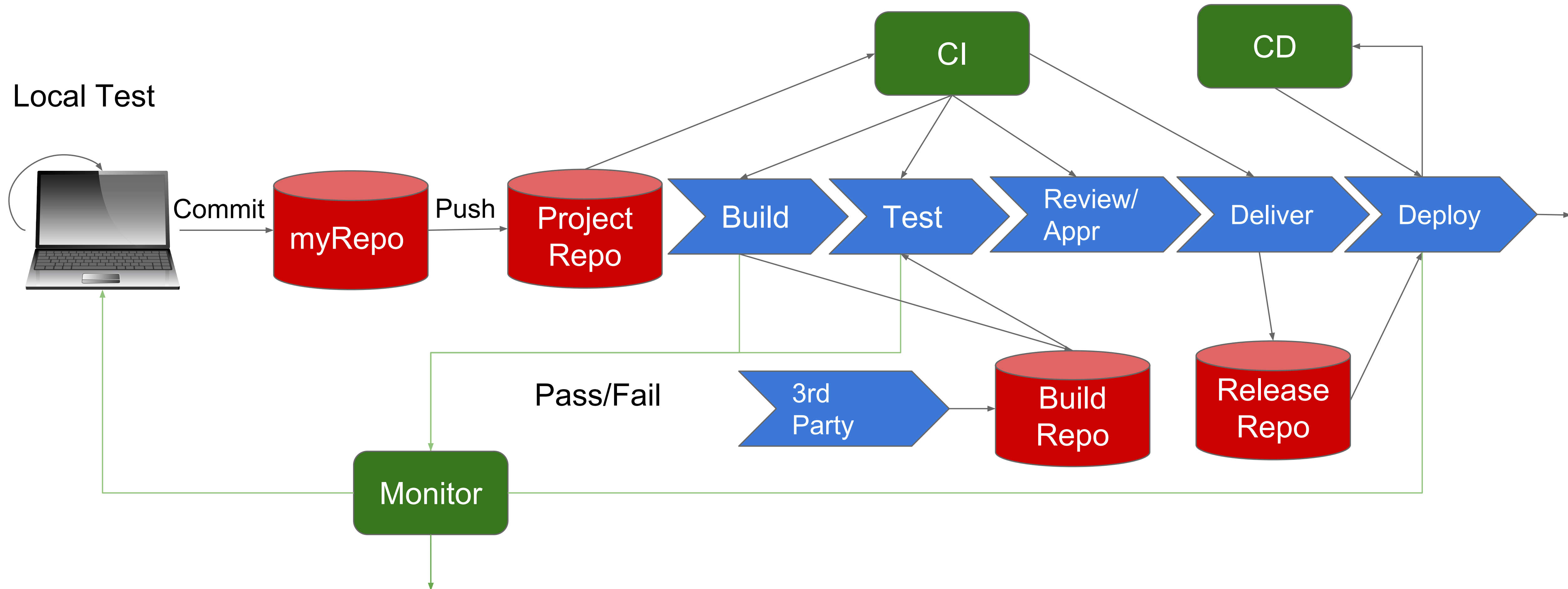
CI/CD Pipeline Toolset



Automation of **PROCESS** from development through ongoing operations

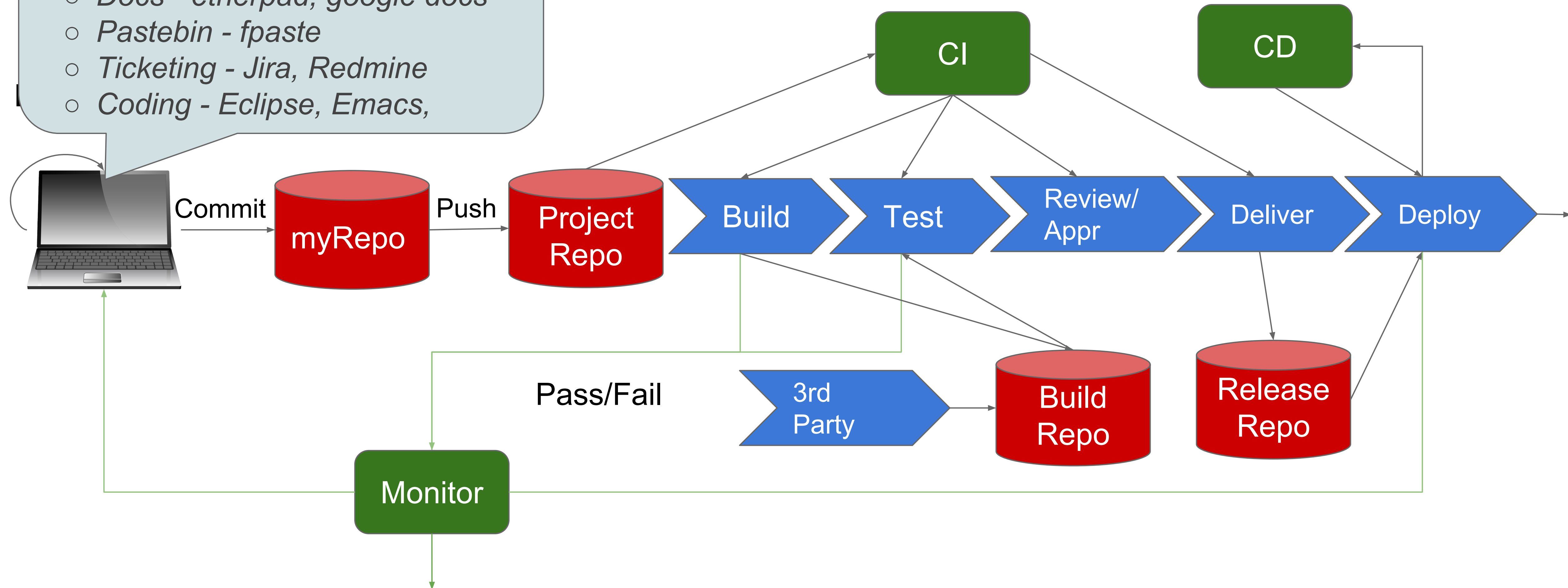


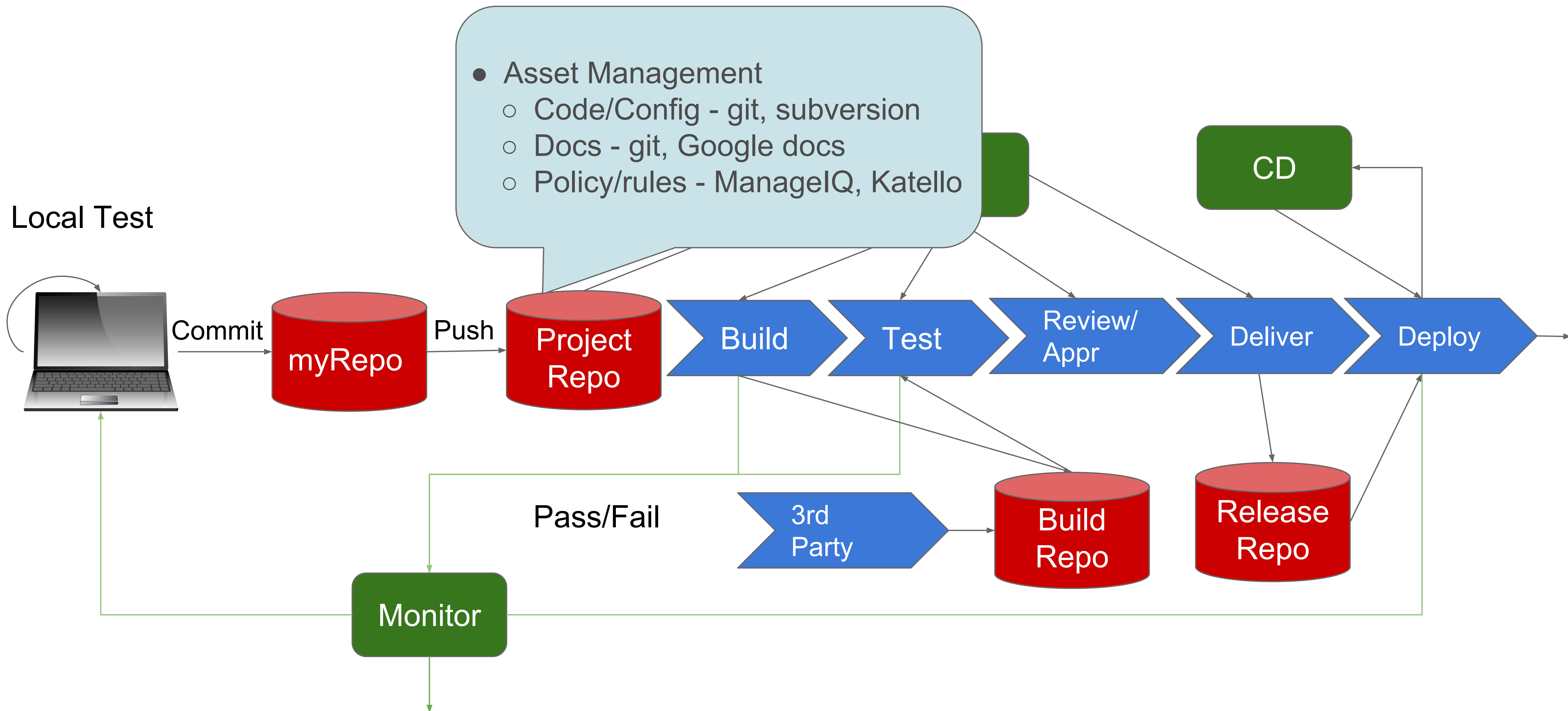
Automation as a CI/CD Process



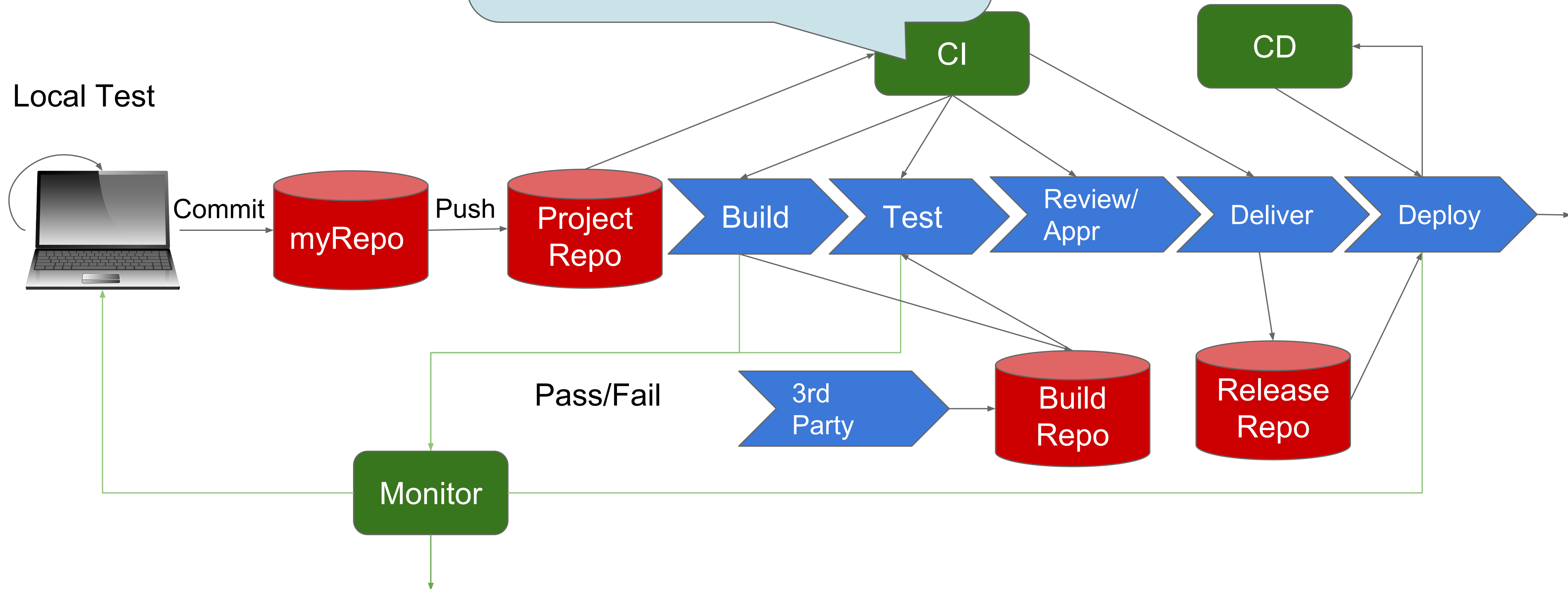
- Collaboration

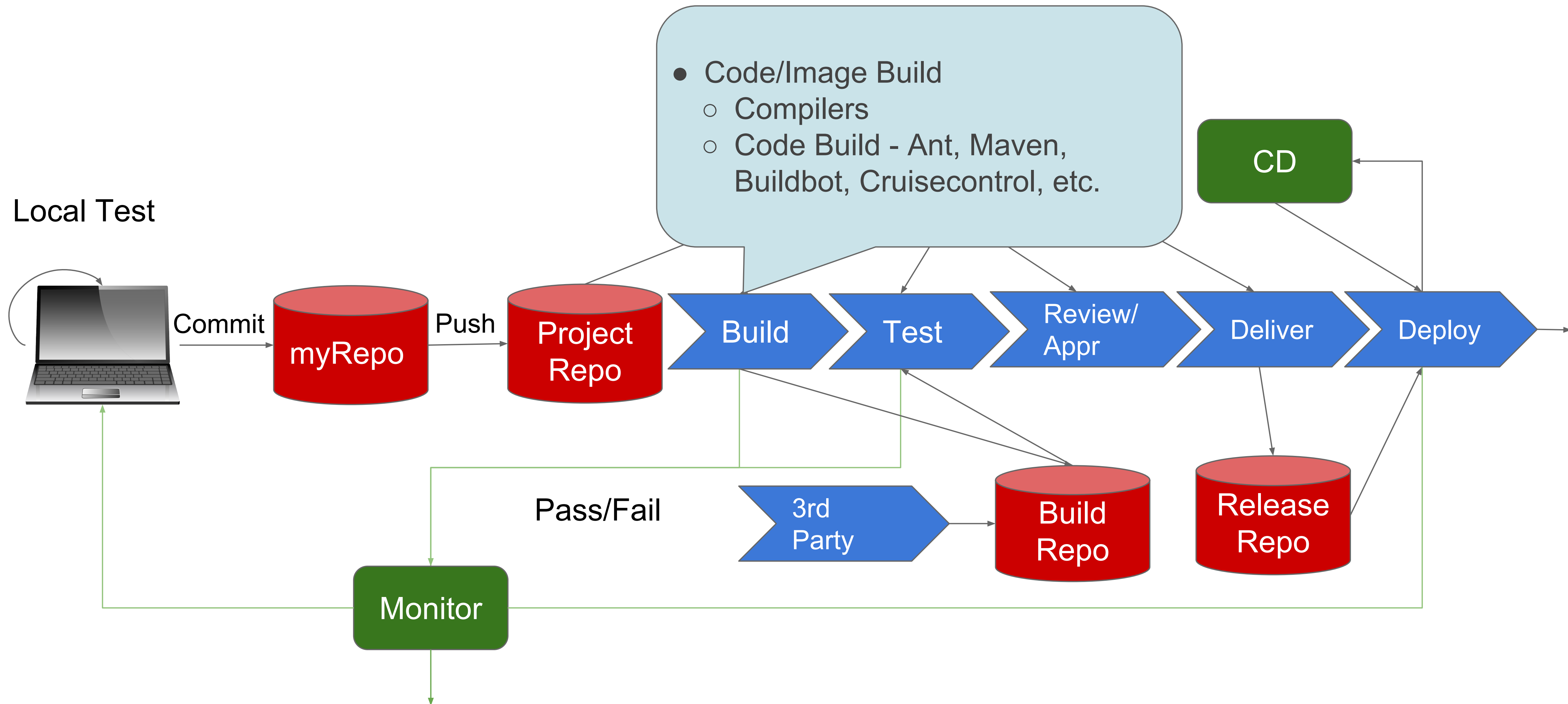
- Kanban - Trello
- Docs - etherpad, google docs
- Pastebin - fpaste
- Ticketing - Jira, Redmine
- Coding - Eclipse, Emacs,

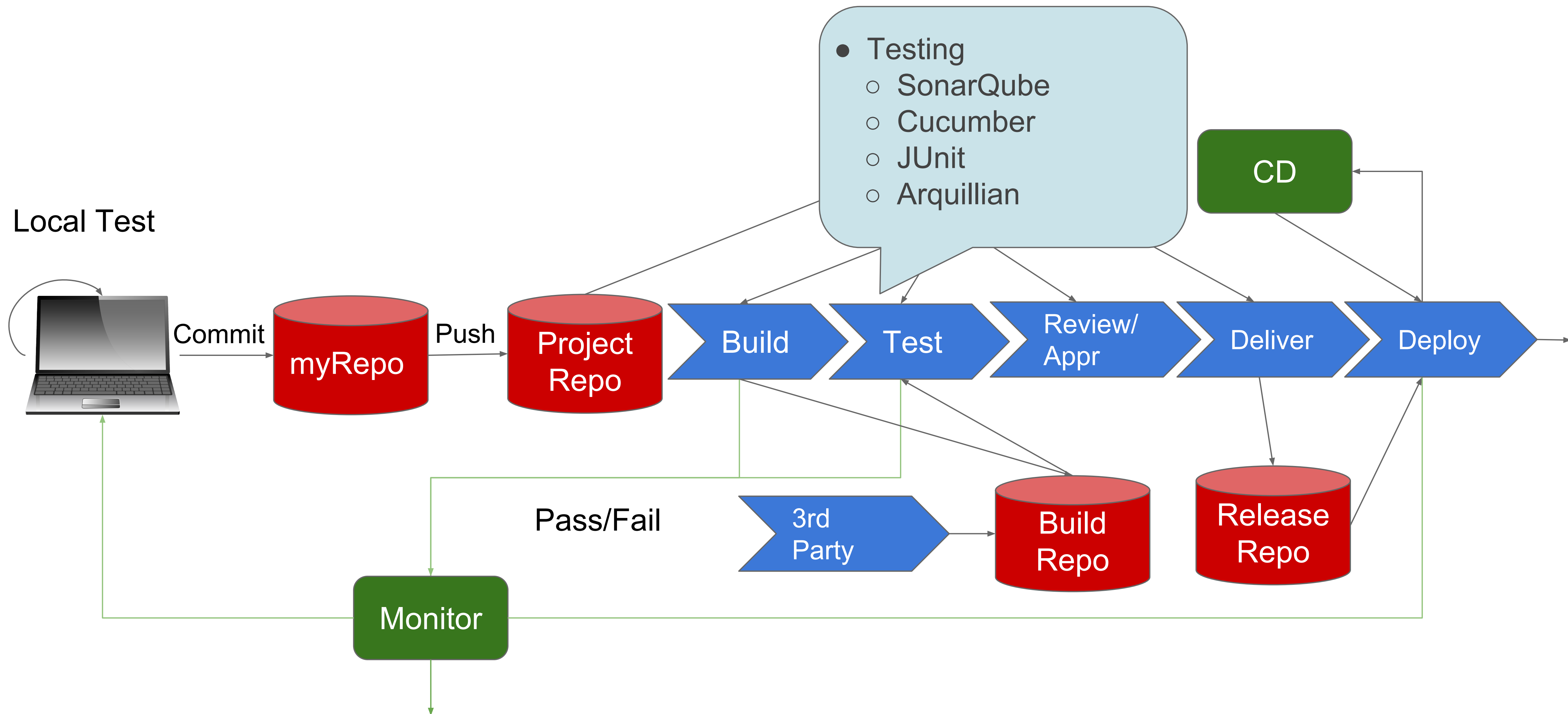


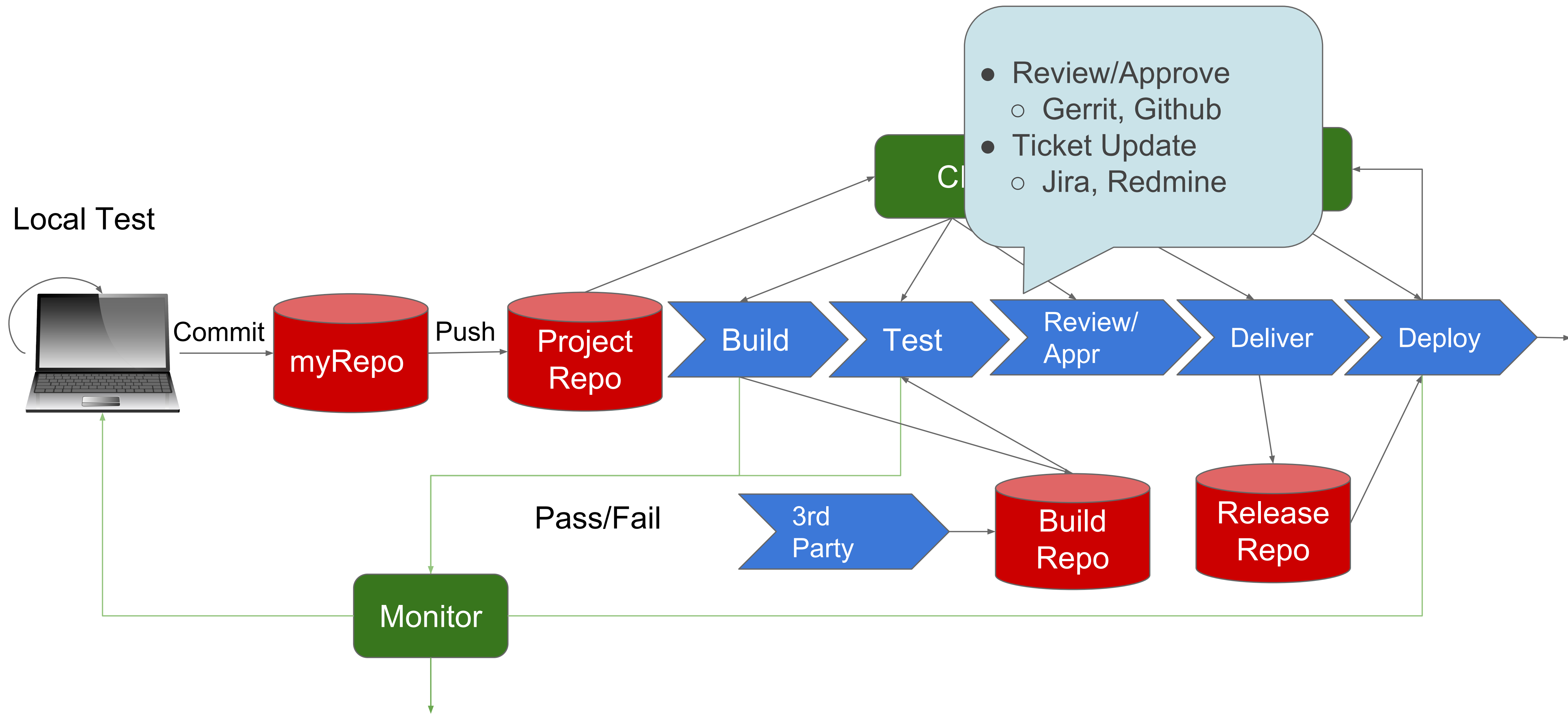


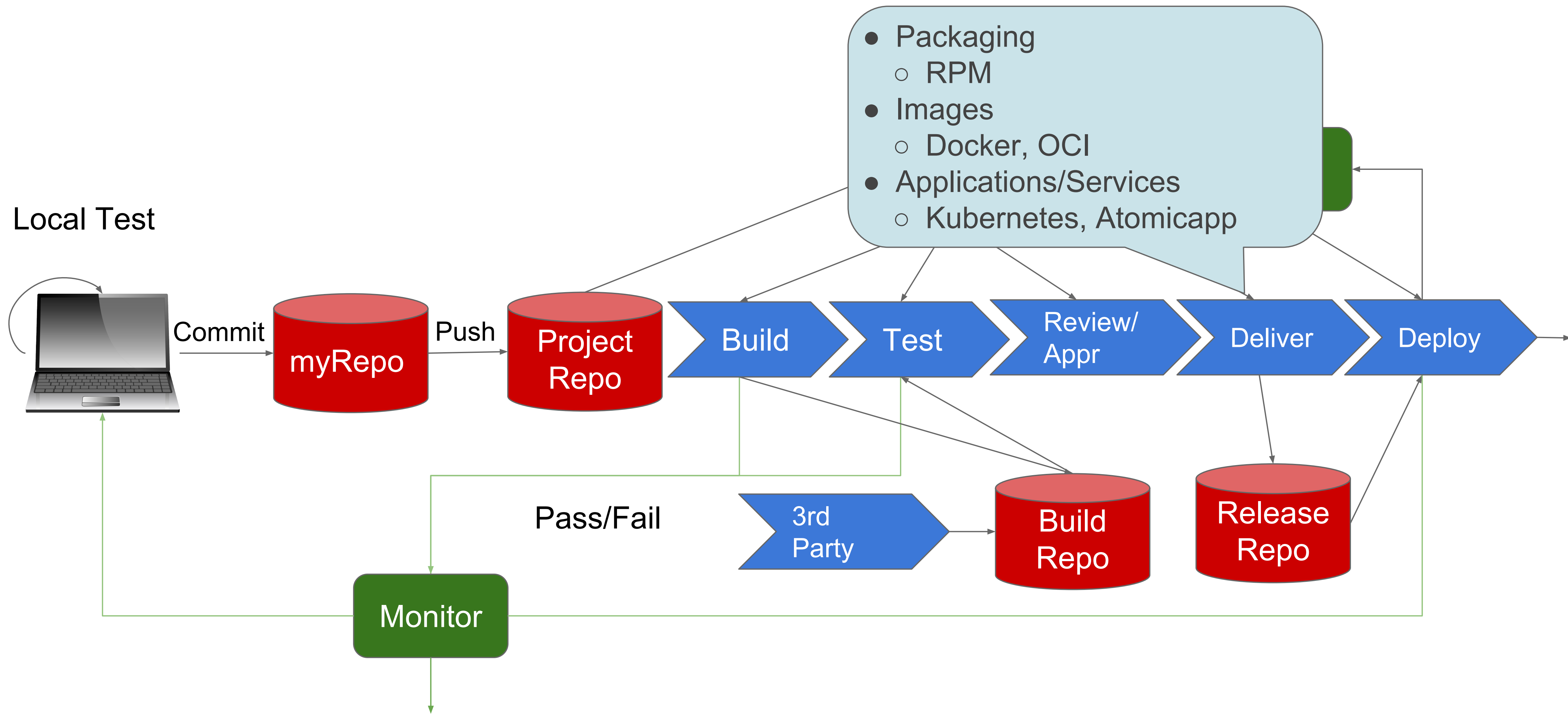
- Continuous Integration
 - Jenkins
 - Travis CI
 - BuildBot

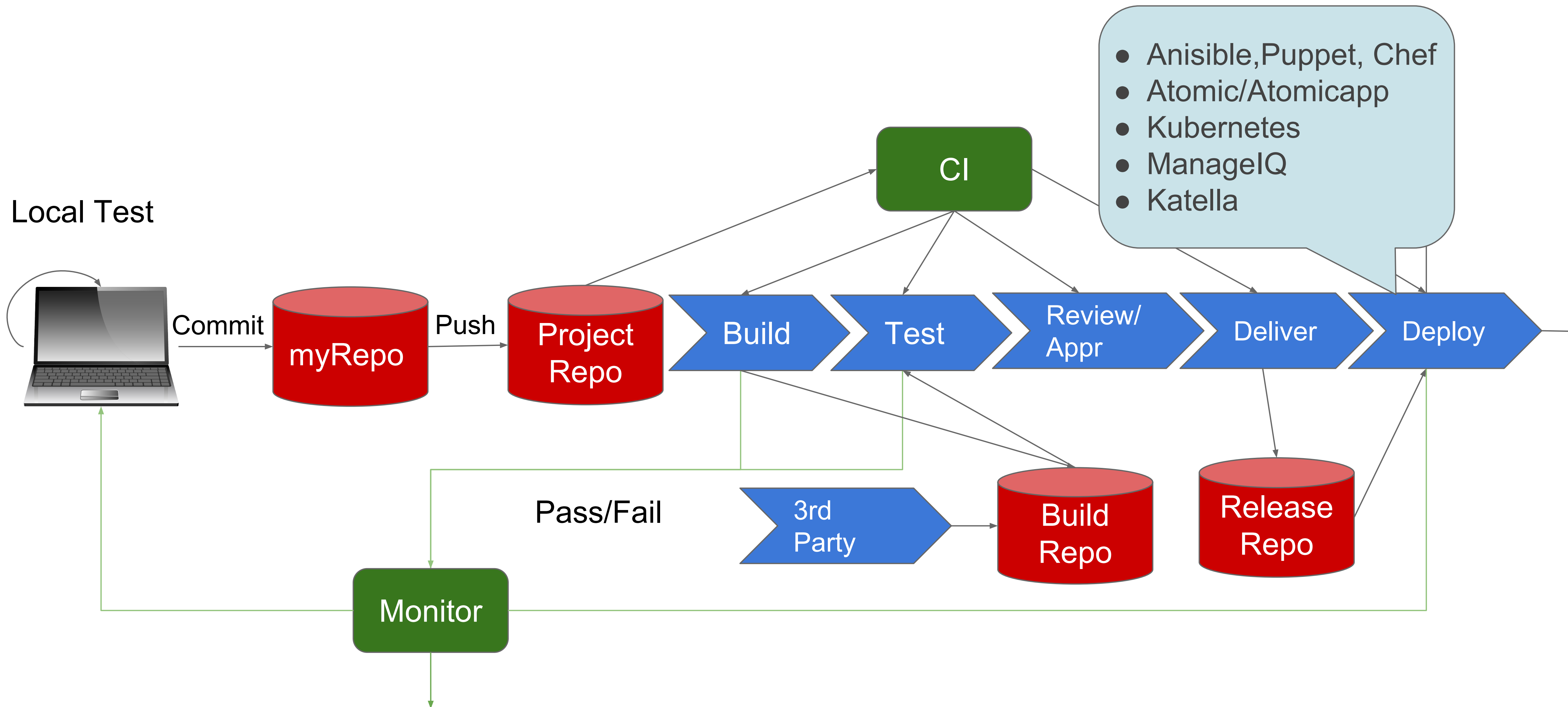


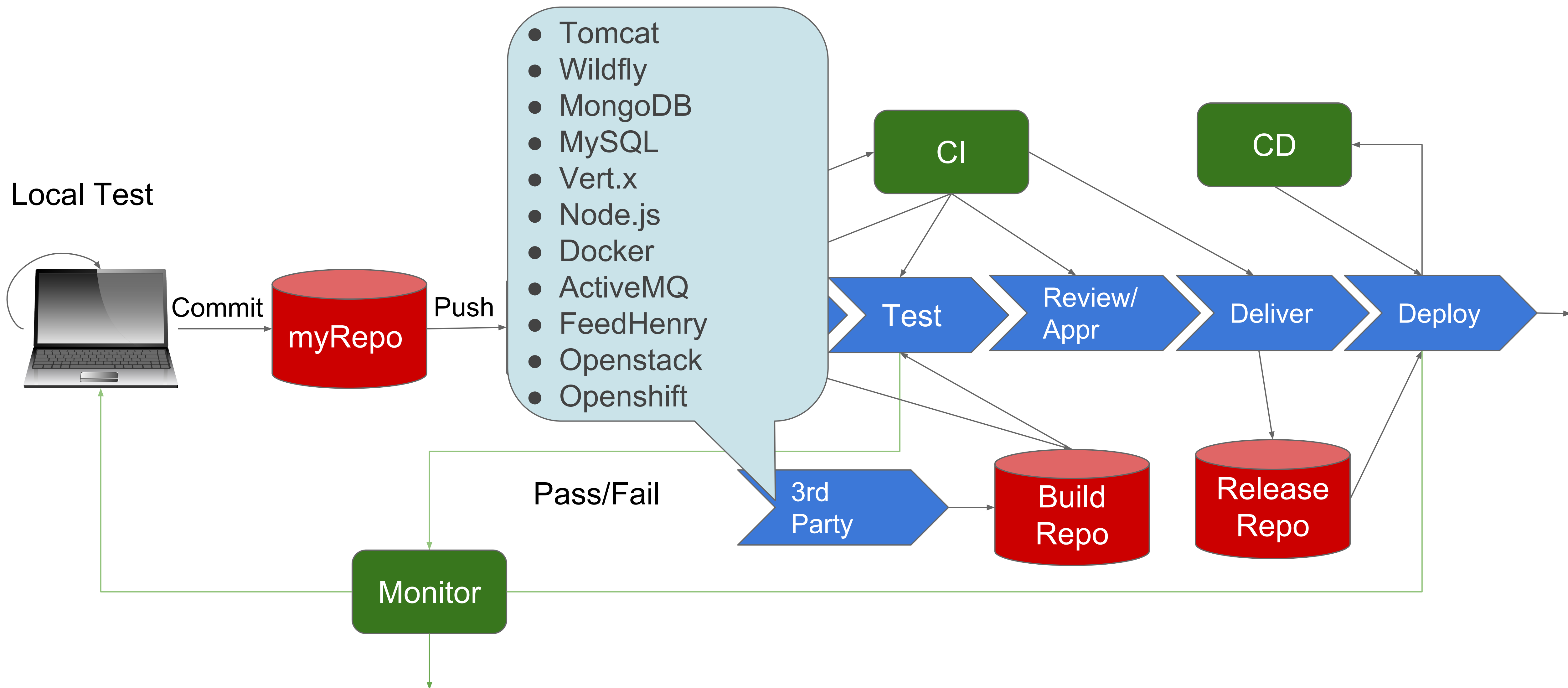




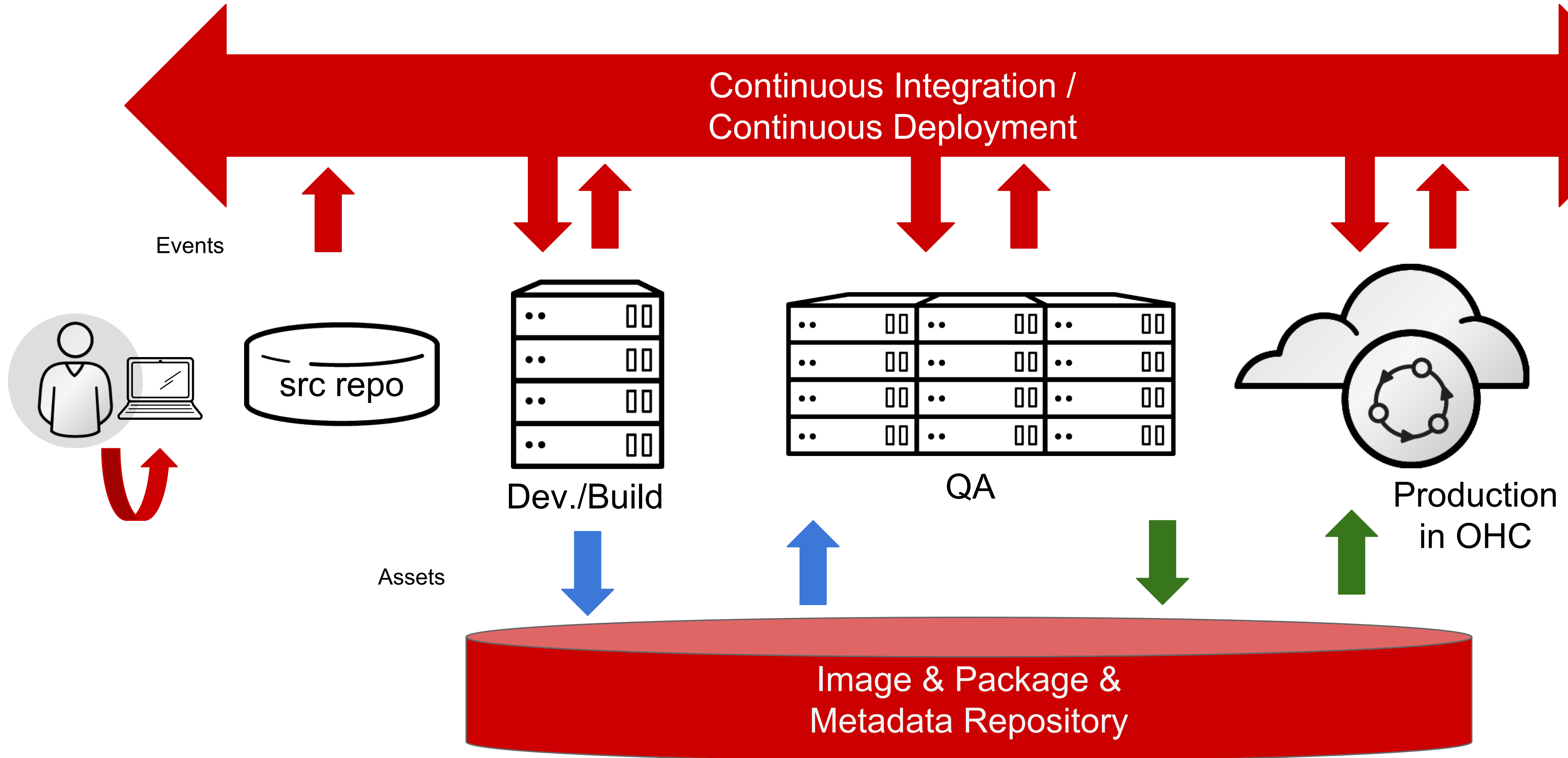




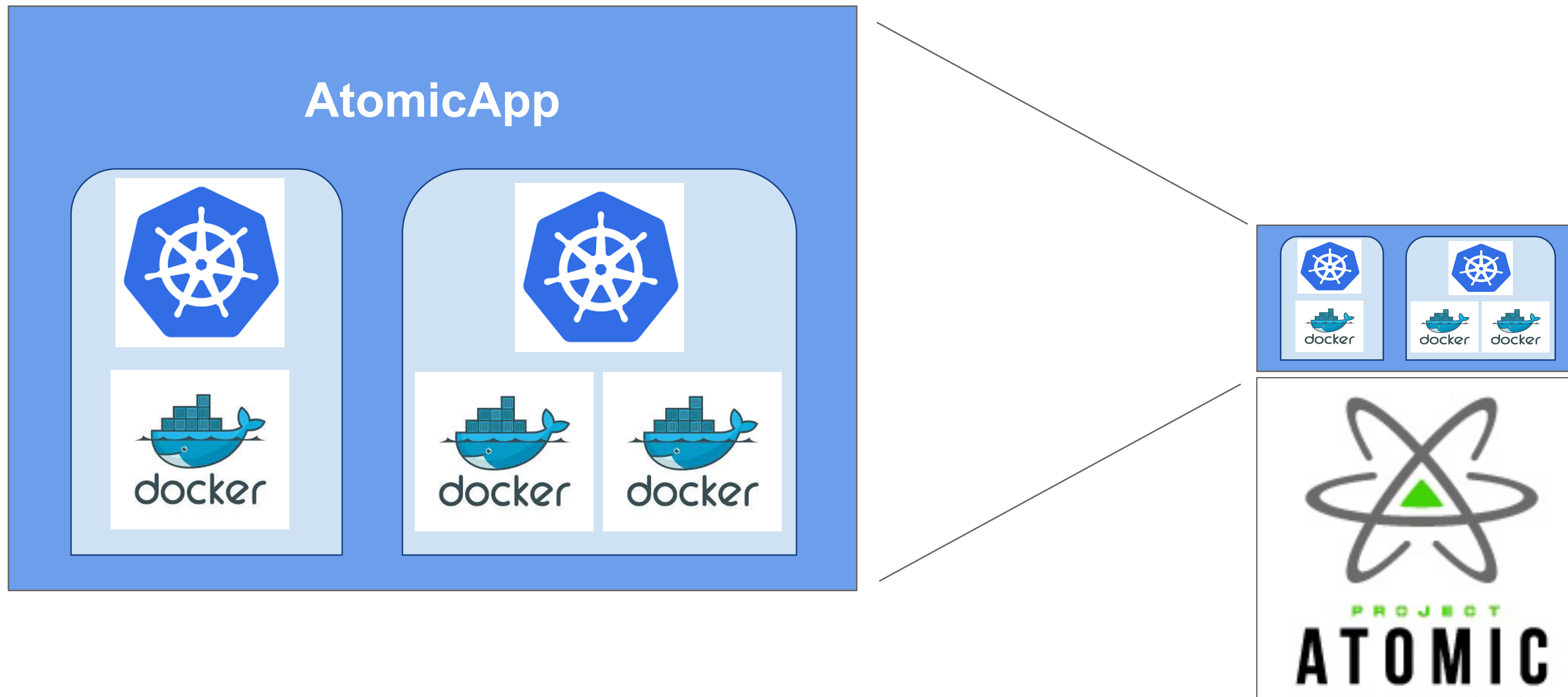




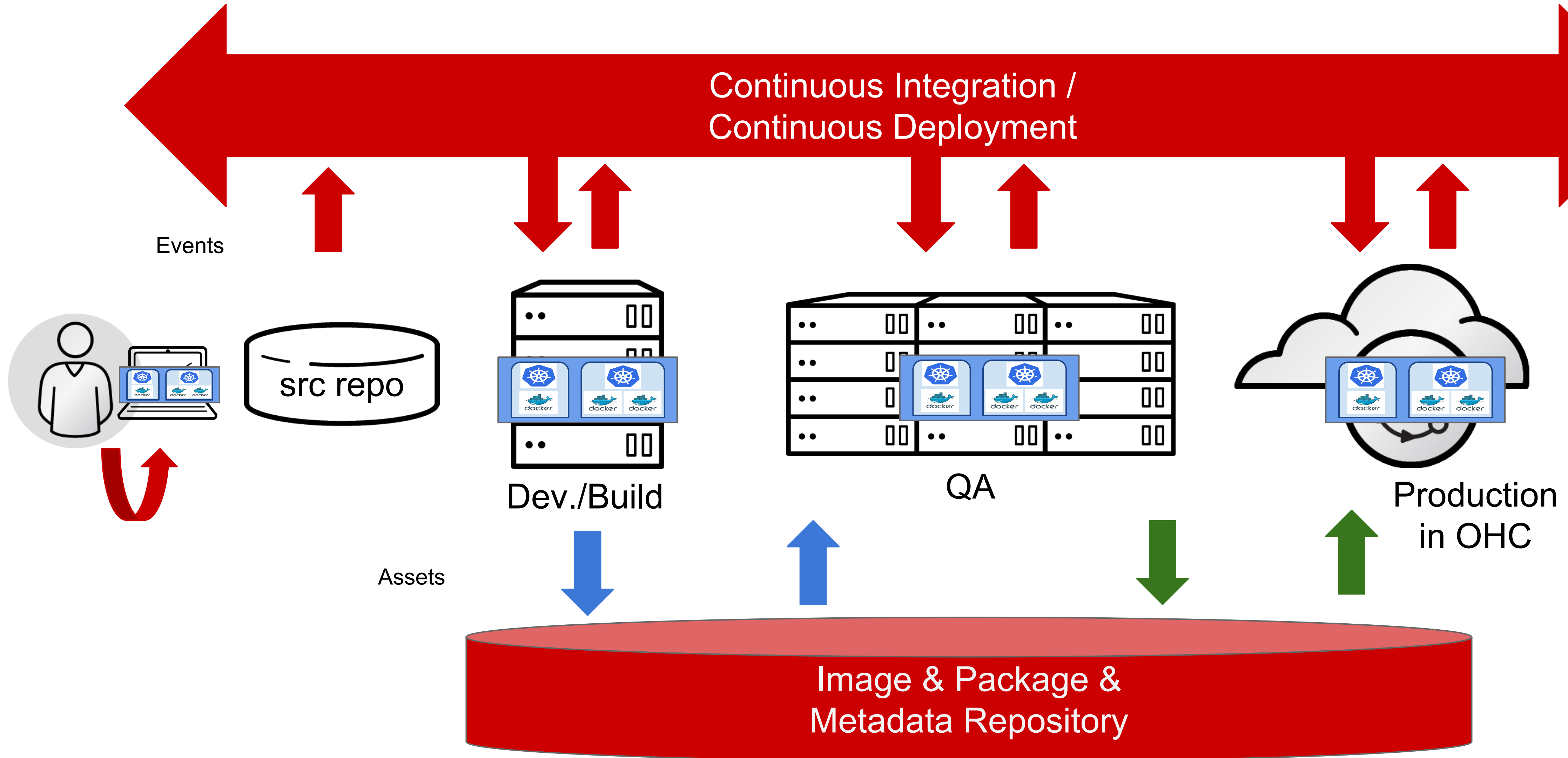
Automation Across Environments



Portable Application



Automation Across Environments



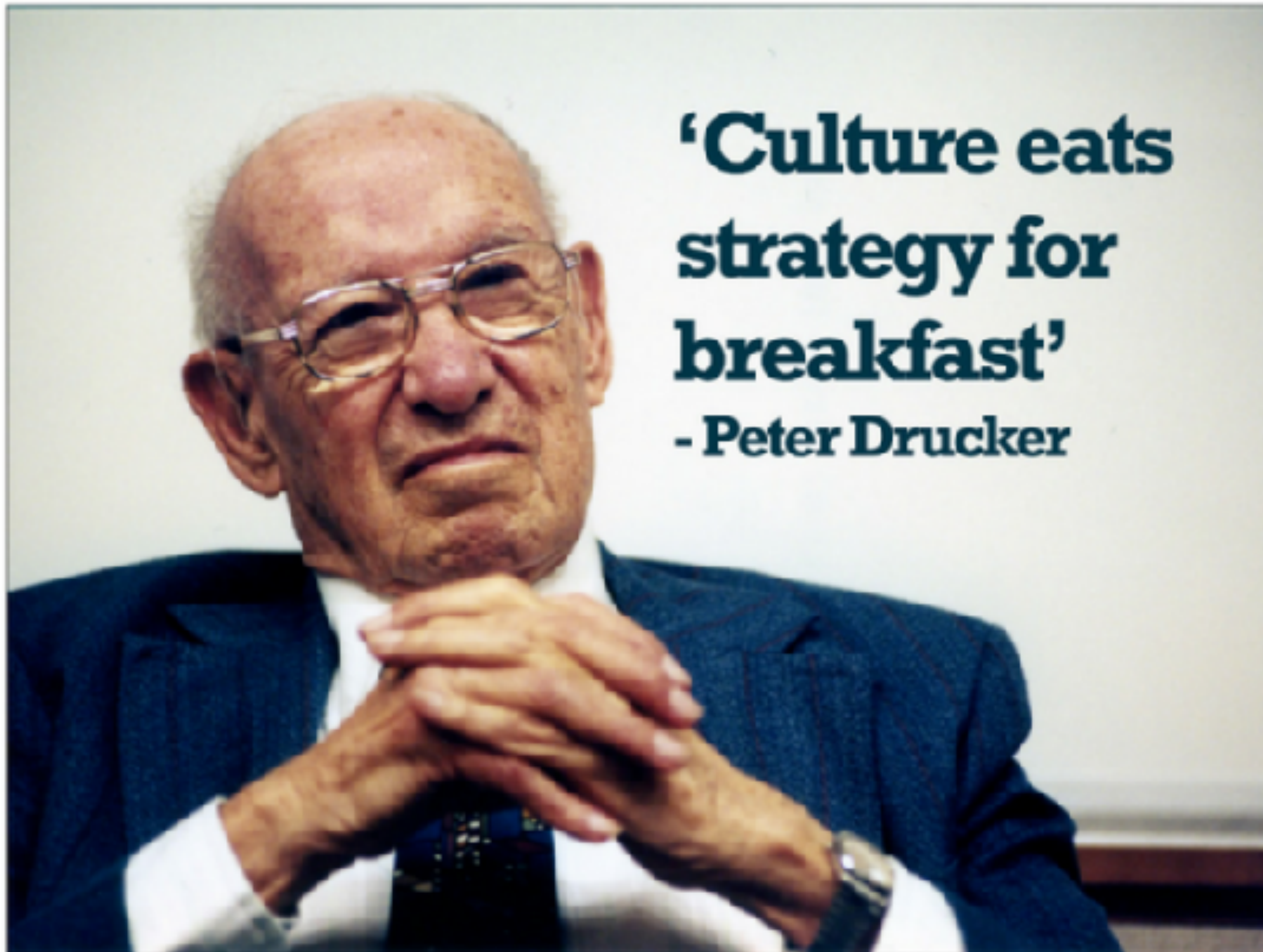
PaaS provides an integration point



CULTURE

**of collaboration
valuing openness
and transparency**

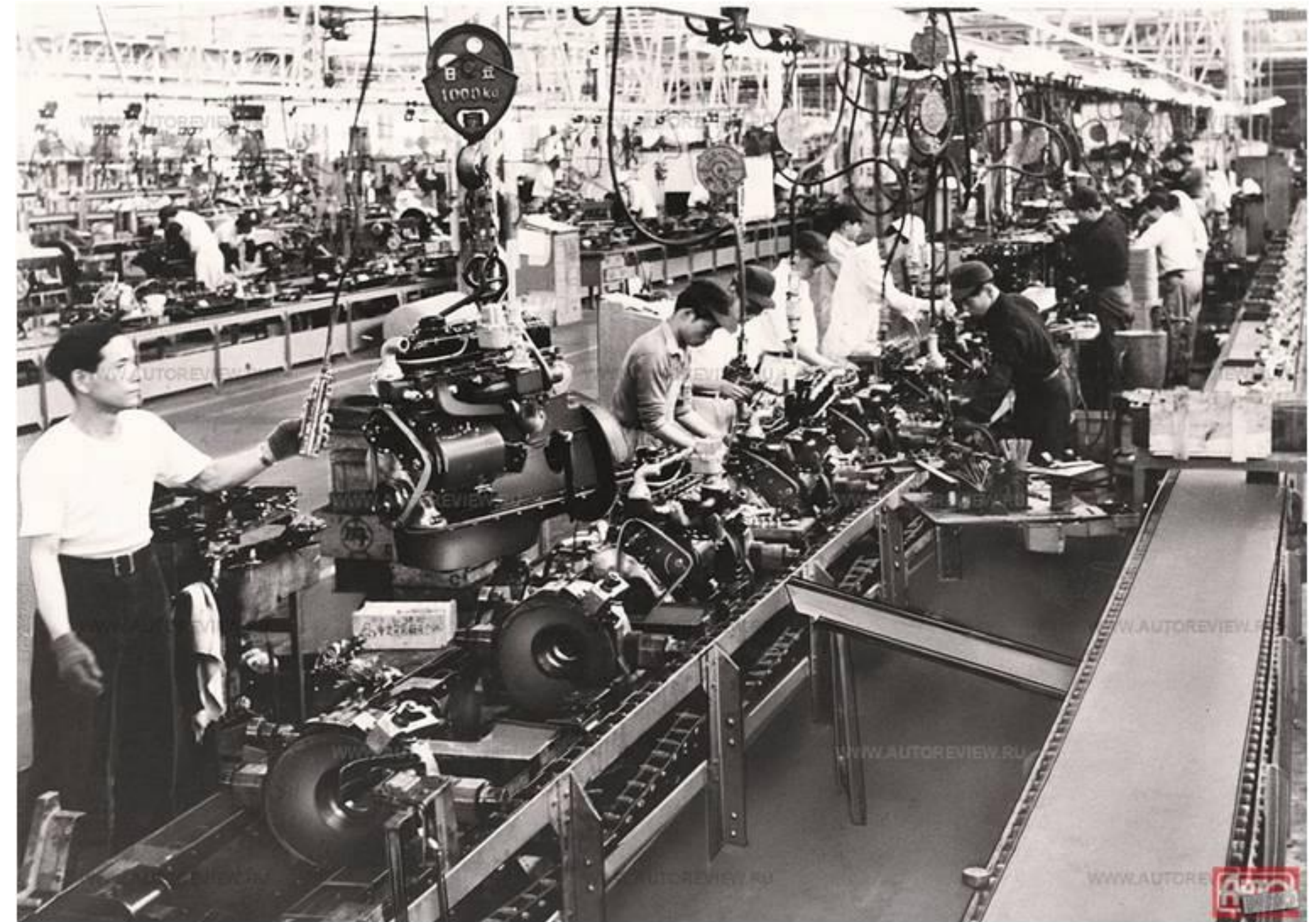




Our culture is too difficult!

Really?

Consider post war Japan.



<https://leanhomebuilding.wordpress.com/page/4/>



Unicorns fart rainbows!

Impose a culture of:
Empathy
Trust
Learning
Cooperation
Responsibility
?



Issues with just “changing culture”

- Lack of agreed-to model of what “right” culture looks like
- Different organizations require different behaviors
- Culture change is difficult to measure and quantify
- Culture is very hard to impose
- Culture is an output, not an input

Culture = f (l, o, i, t, ...)

where:

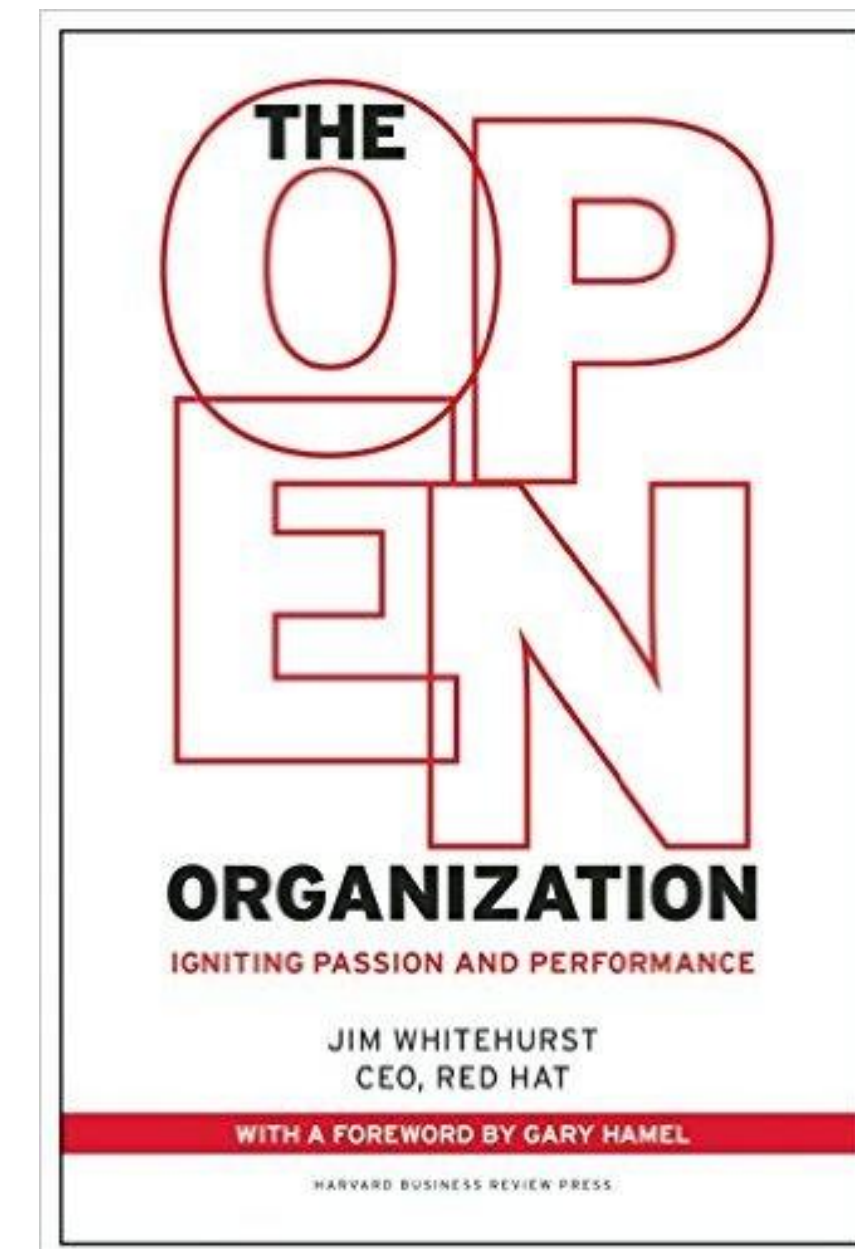
l = leadership

o = organization

i = incentives

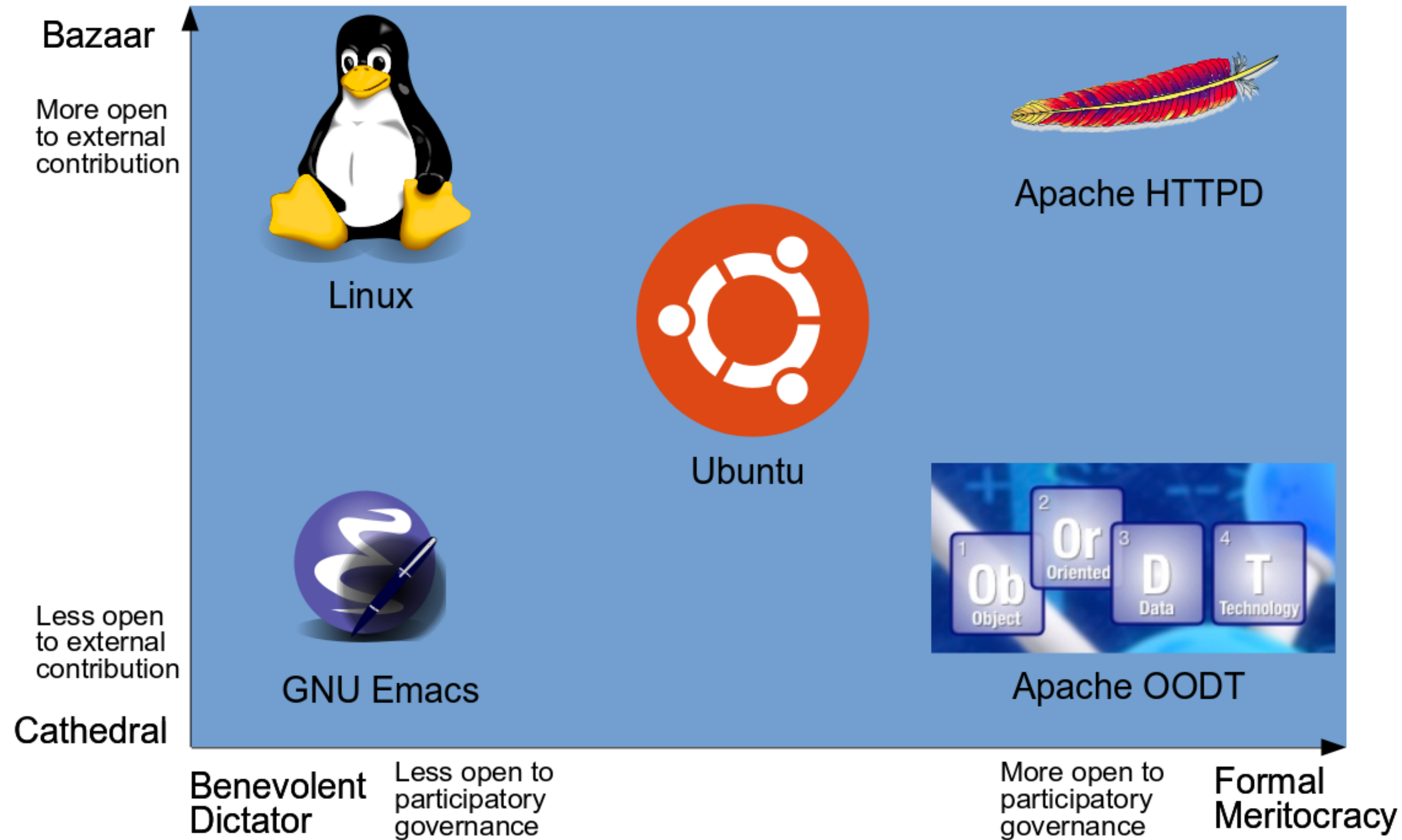
t = trust

... = many other things



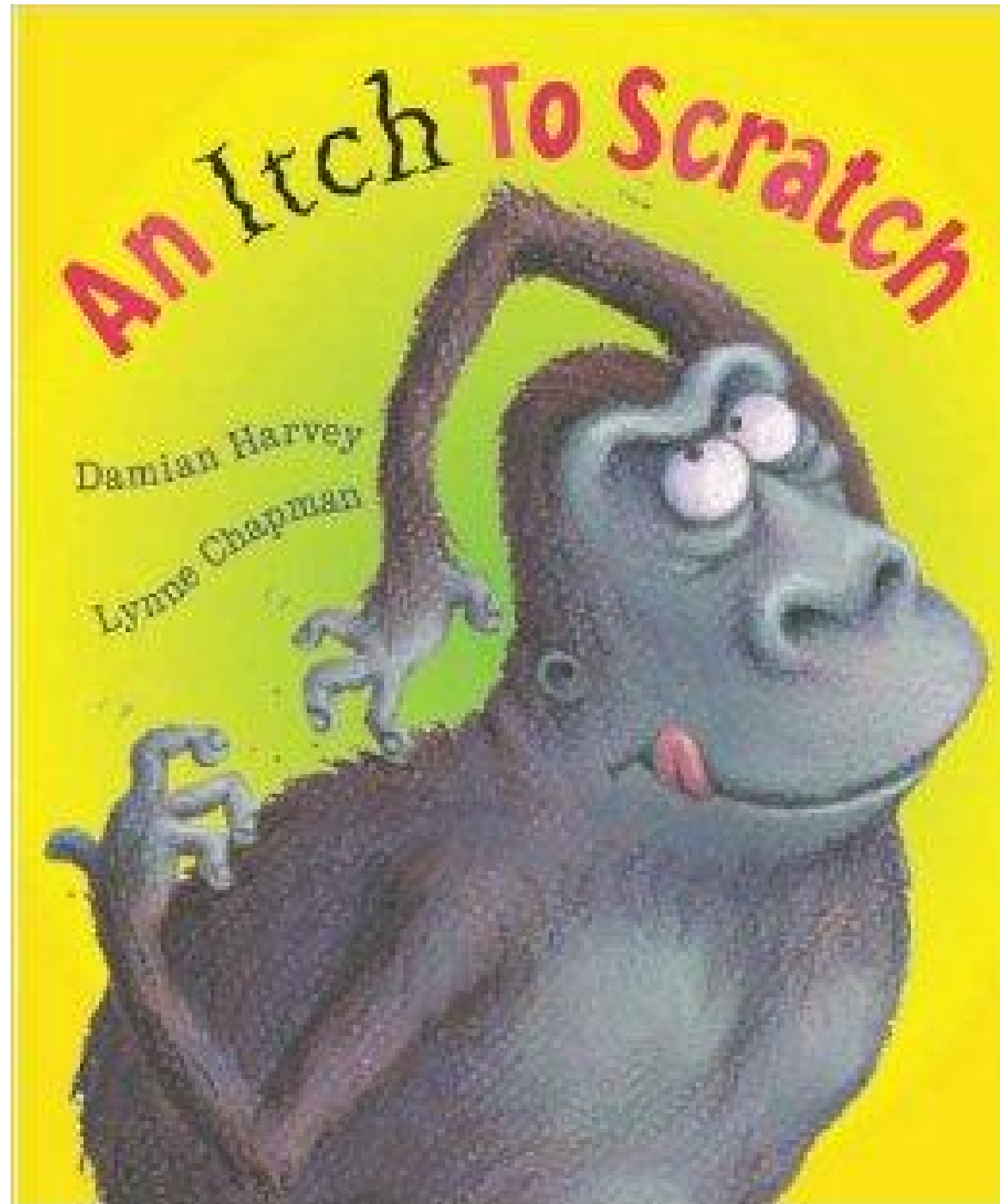
Open source offers guidance

Leadership and vision but no one organizational structure



Multiple models work but characteristics need to fit the project, participants, and vision

Solve specific problems



Open source projects often begun because of an individual's interest or issue

DevOps projects (often) benefit by tackling low surface area problem cases

(But, like open source, DevOps can become the default)

Transparency



Who made changes?

When and why did they make them?

What's the state of the project?

What's the state of the system?

It's the expectation for both open source and DevOps

Rich communication flows matter



Open source projects have been forced to deal with distributed (cross-timezone) teams

Tools vary by teams

Video getting better

F2F time still a big plus

Don't fear failure



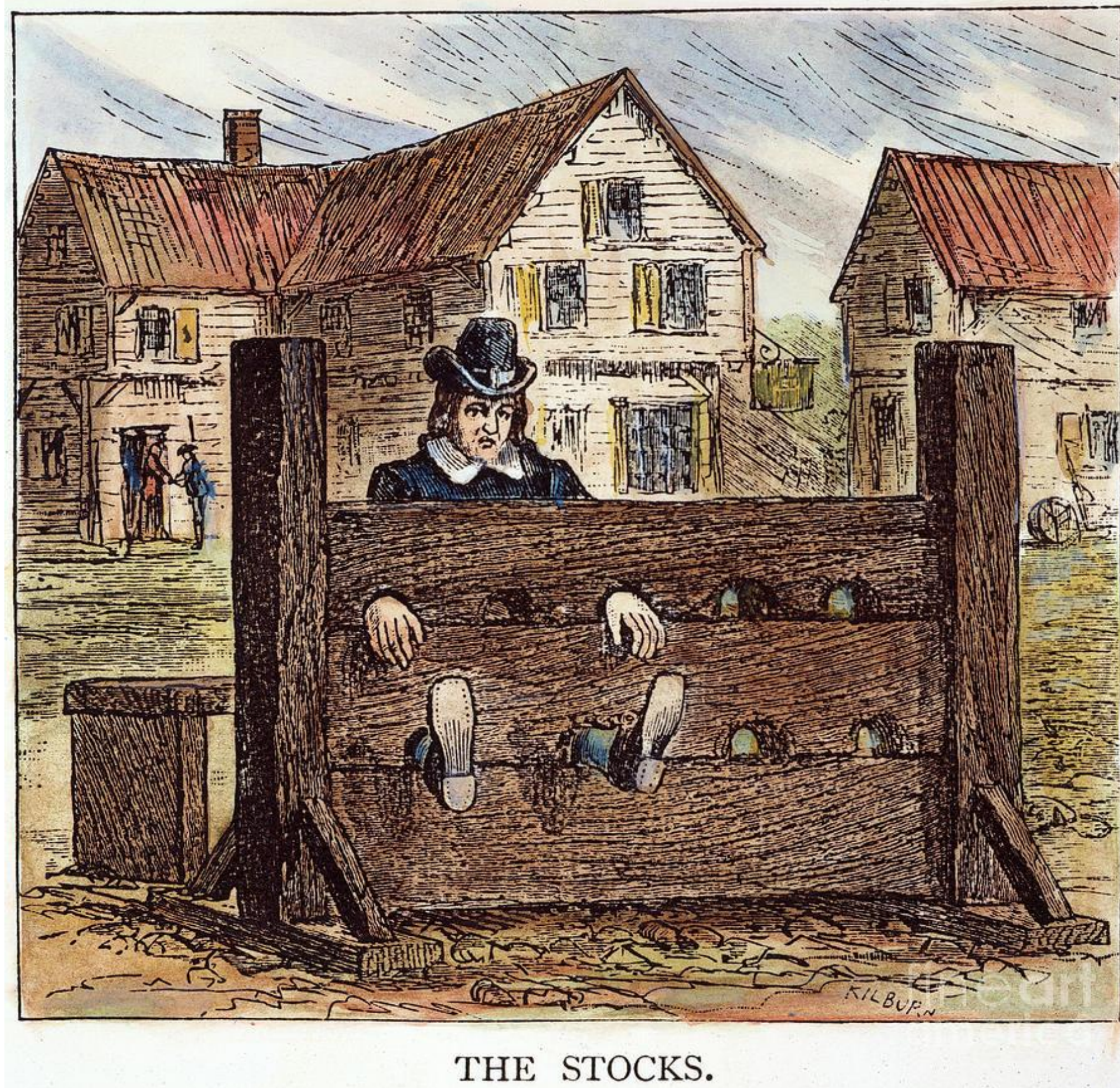
Open source innovation highly driven by experimentation

One of the points of DevOps is to enable better experimentation

If it doesn't work, move on

(But fast failures)

Incentives matter



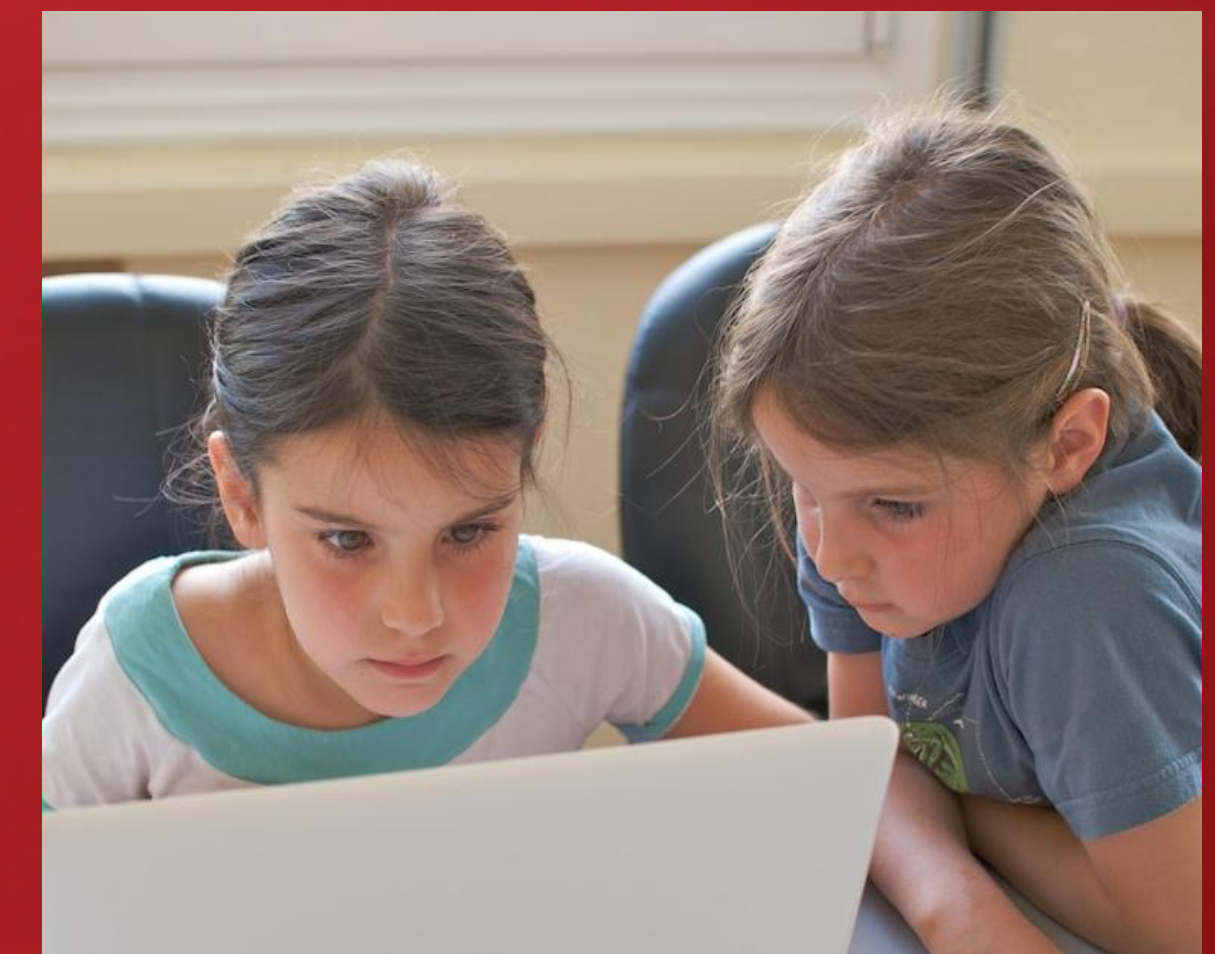
Open source projects (tend to) reward based on merit and contribution

Incentives in DevOps organization (advancement, money, recognition) need to reward trust and cooperation

Individual has control over their own success

Open source is driving DevOps

But it's more than the code!



Credits

Tools: Dorli Photography, cc/flickr <https://www.flickr.com/photos/dorlino/4946061042/>

Robots: davidgariepy, cc/flickr <https://www.flickr.com/photos/davidgariepy/2495011427/>

Kids programming: Esti Alvarez cc license

Dev: Nelson Pavlosky/flickr under CC <http://www.flickr.com/photos/skyfaller/113796919/>

Ops: Leonardo Rizzi/flickr under CC <http://www.flickr.com/photos/stars6/4381851322/>

Rainbows and Unicorns: <http://kaigumo.deviantart.com/art/Unicorns-Fart-Rainbows-3-151273843>

Join hands: <https://www.flickr.com/photos/vogelium/10565496565/>

Thanks to J.P.Morgenthal for some of his thoughts about culture change. See also <http://www.slideshare.net/JPMorgenthal1/process-andorgtrumpcultureinentdevops>

Governance models: <http://oss-watch.ac.uk/resources/governancemodels>

Transparent code: iStockPhoto

Linux Collaboration Summit: Linux Foundation

Wipeout: <https://www.flickr.com/photos/andymorffew/15843725192>

No: Wikimedia

Thank you!

Questions?