redhat.

Operations Workflow for Customizing Builder Images

By: Karl Moos, Nicholas Poyant and Frank Kannemann

Agenda

- Historical OPS Discussion
- Overview of the NEW CICD/DEVOPS workflow
- Technical Deep Dive discussion
- Demo using “Ticket Monster” sample application

Note: This Presentation does not incorporate the the next gen of Cloud Management with Cloud Forms (Aka ManageIQ).

Historical OPS Discussion

Historical OPS Roles

Here are the generally agreed upon roles of OPS:

- Build & Deploy
- Scheduled Maintenance
- Security/Vulnerability Mitigation
- Monitoring

This presentation is based on the concept of an new additional “Operations Environment” (Which today is probably a lab).

Segregation of Roles in the Old Workflow

Most of workflow approaches for CI/CD are Development based

- There are 3 equally important roles in today's traditional I.T. Shops
  - Developers
  - QA
  - Operations
- This 3 step workflow is starting to blend together into DevOps
- One term describing this methodology is: Waterfall/ITIL

DEVOPS Integration with OSE

DEVOPS

The definition of DEVOPS from Wikipedia:

"....is a culture, movement or practice that emphasizes the collaboration and communication of both software developers and other information-technology (IT) professionals while automating the process of software delivery and infrastructure changes…"

- To support OPS in DEVOPS CI/CD the following will be described and demonstrated:
  - New "Operations Workflow" in a CI/CD environment in
  - How this can be easily implemented OSE

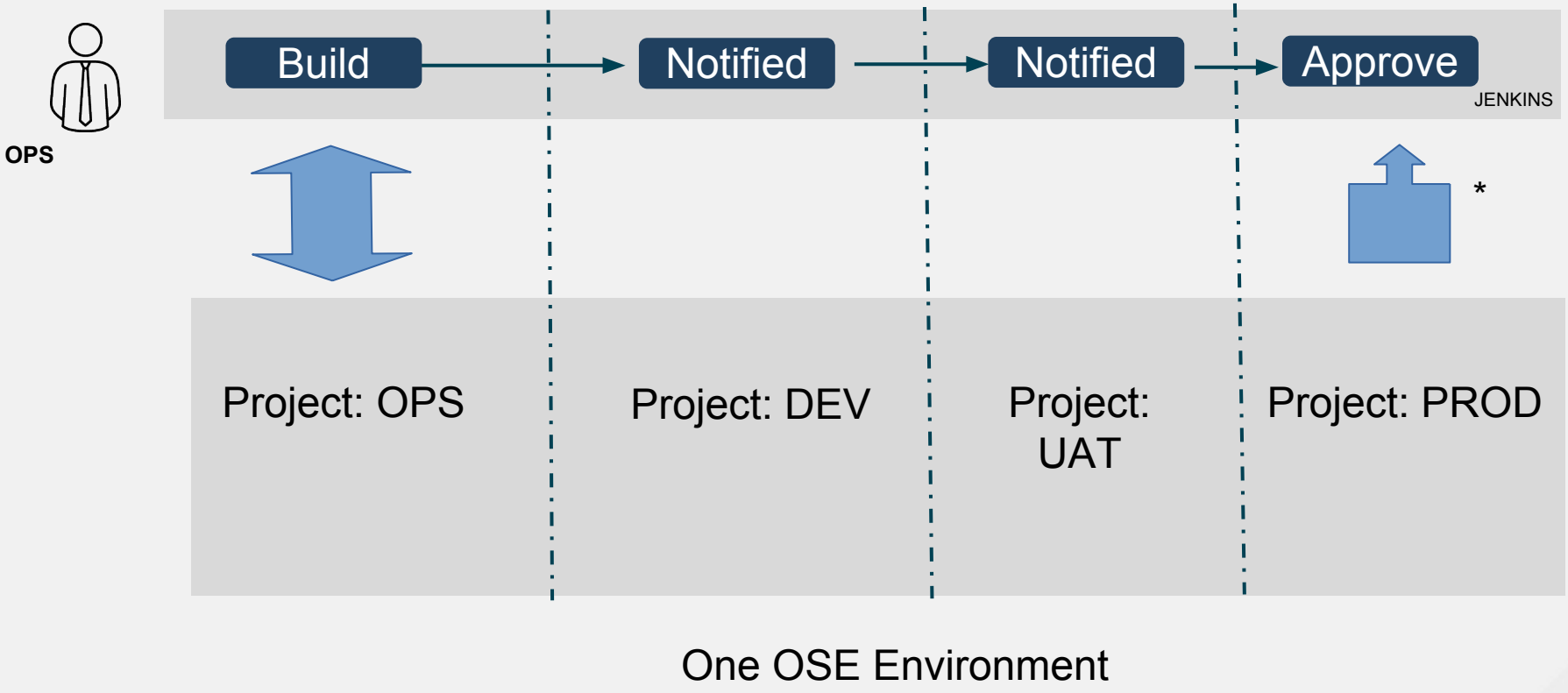
New OPS Role in the OSE CI/CD Workflow

The OPS workflow simplifies the touch points down to 2 (At the beginning and at the end). Most of the work being upfront.

1. *At Start of Project*
   - Builds/Maintenance (patching)
   - Security (see above, access control)
2. *At Deployment to Production*
   - Deployment (to PRODUCTION)
   - Monitoring/Alerting (OMG it's down)

- One term describing this methodology is: Agile/DevOps
- The next slide introduces the new “Operations Workflow” in a CI/CD environment as provided by OSE

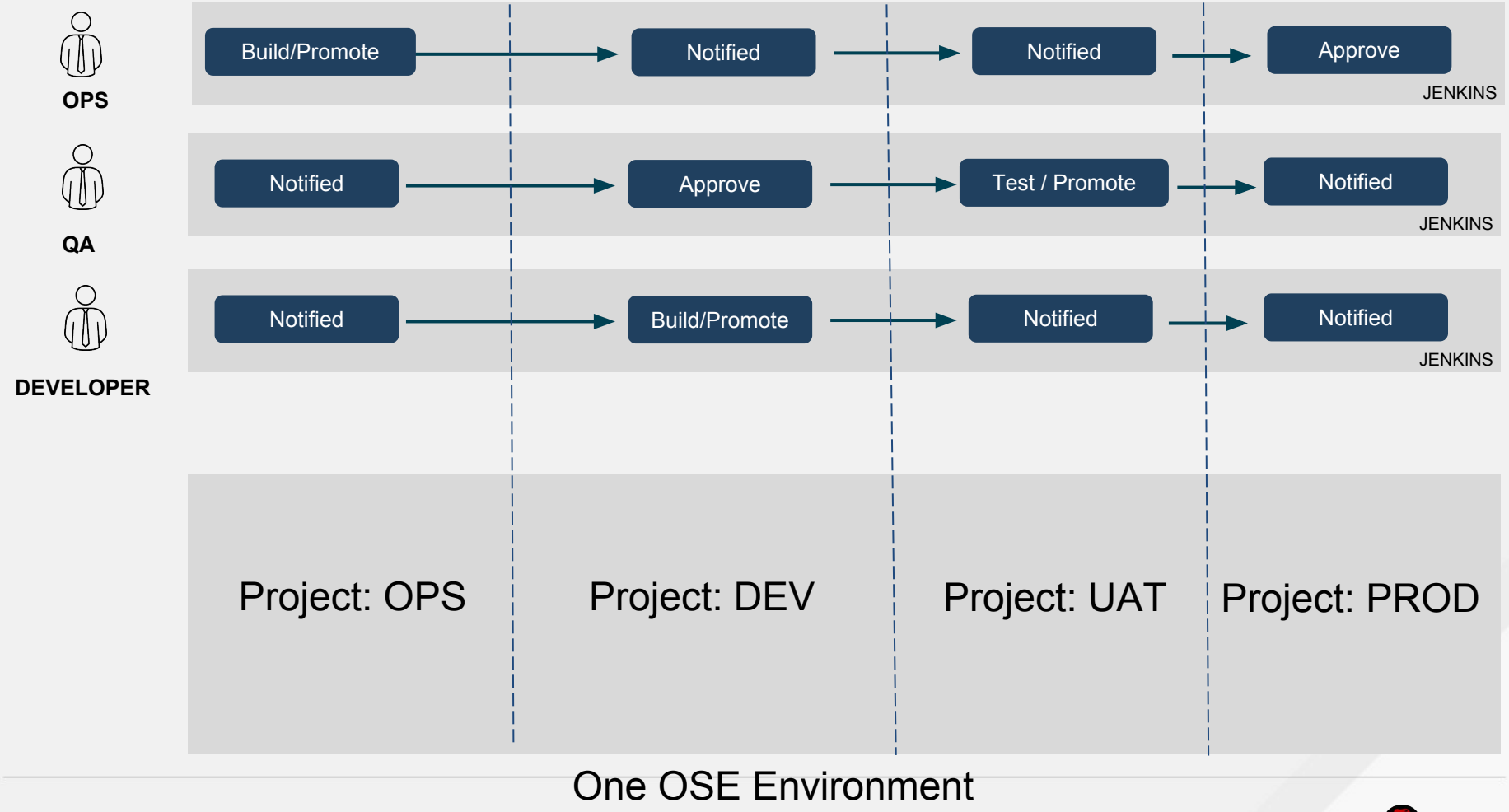
PROPOSED NEW OPS TOUCH POINTS IN CI/CD



Notes::

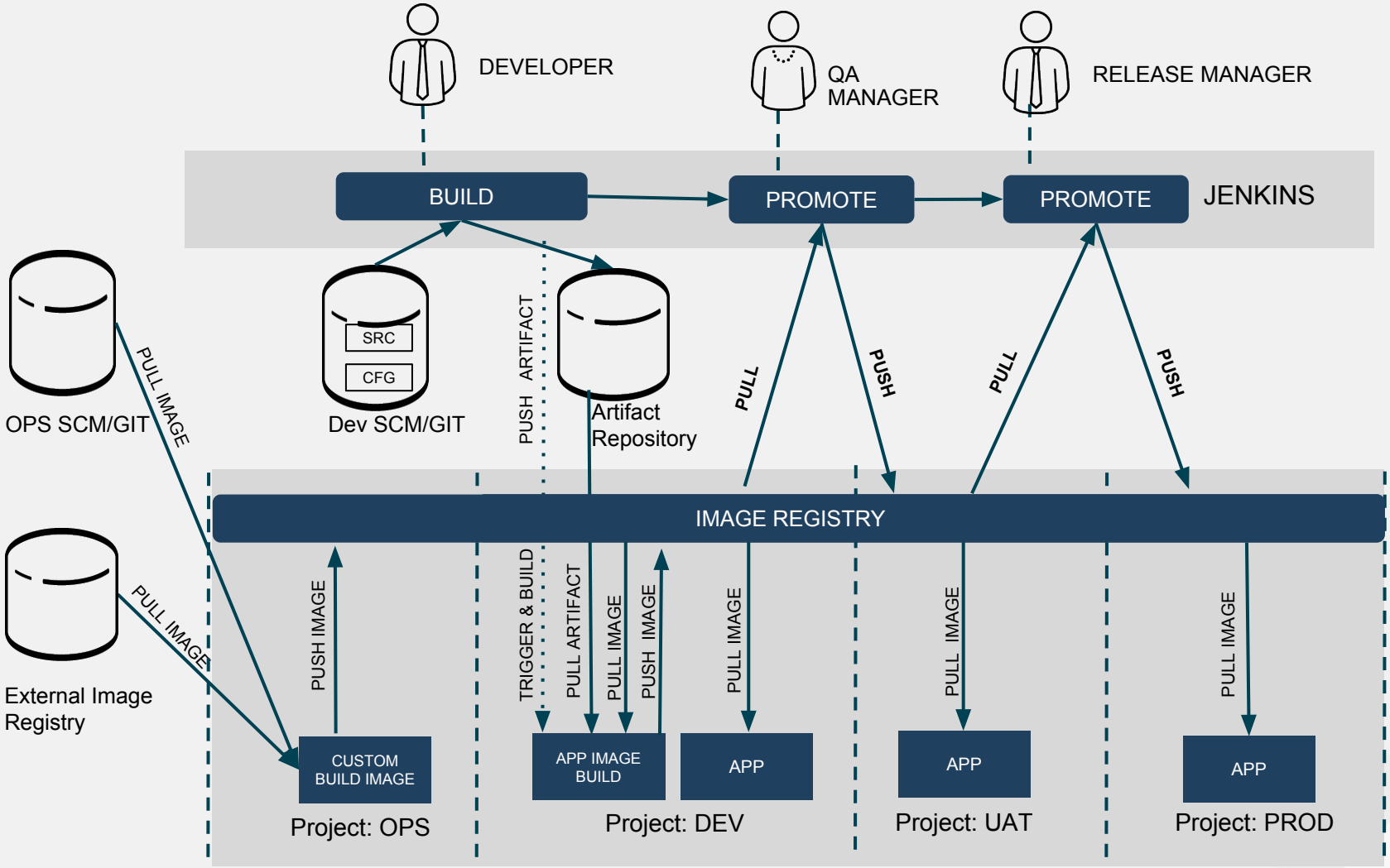
This one OSE environment can split into separate OSE instances as project or organizational policies require.
** This is not standard in CICD*

Holistic View of DEVOPS with CI/CD Using OSE

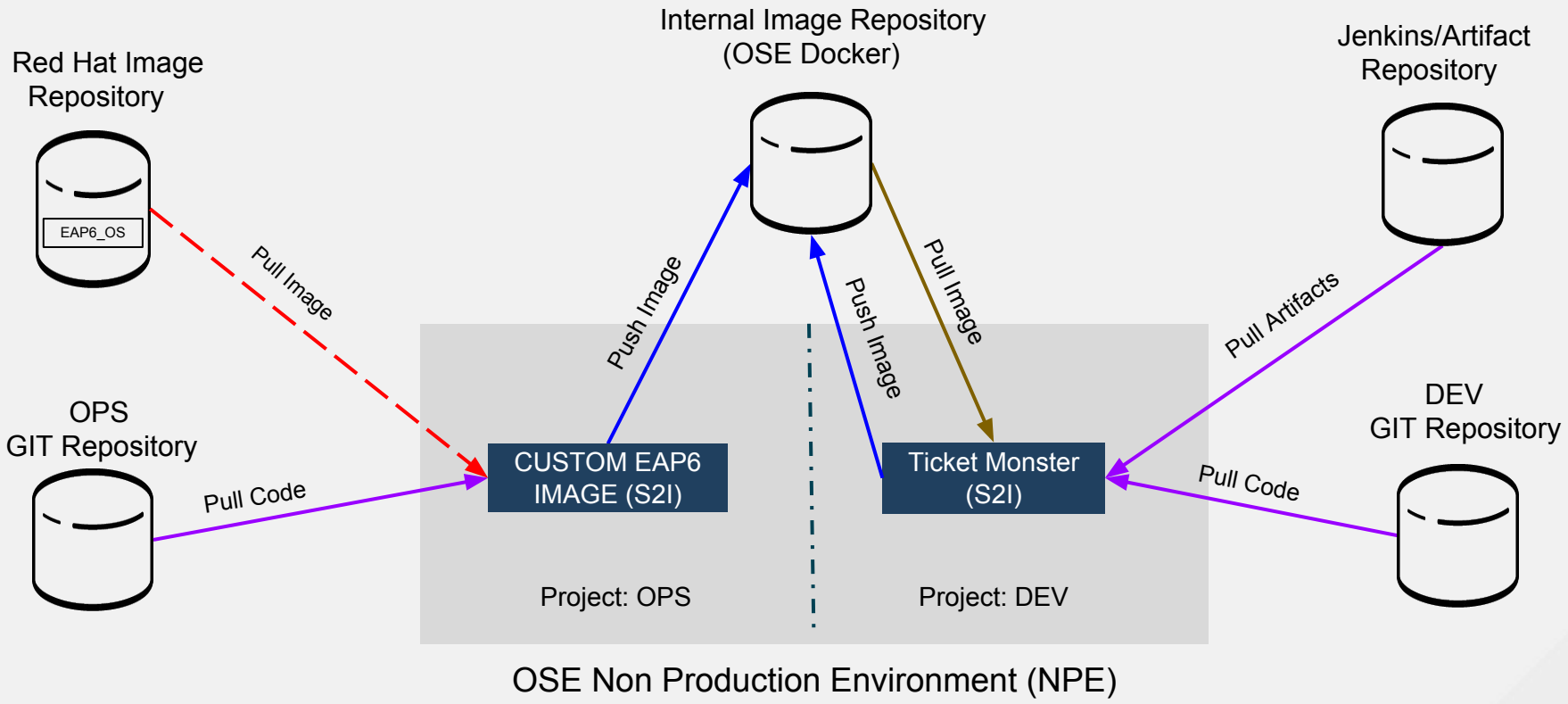



Technical Deep Dive
redhat.

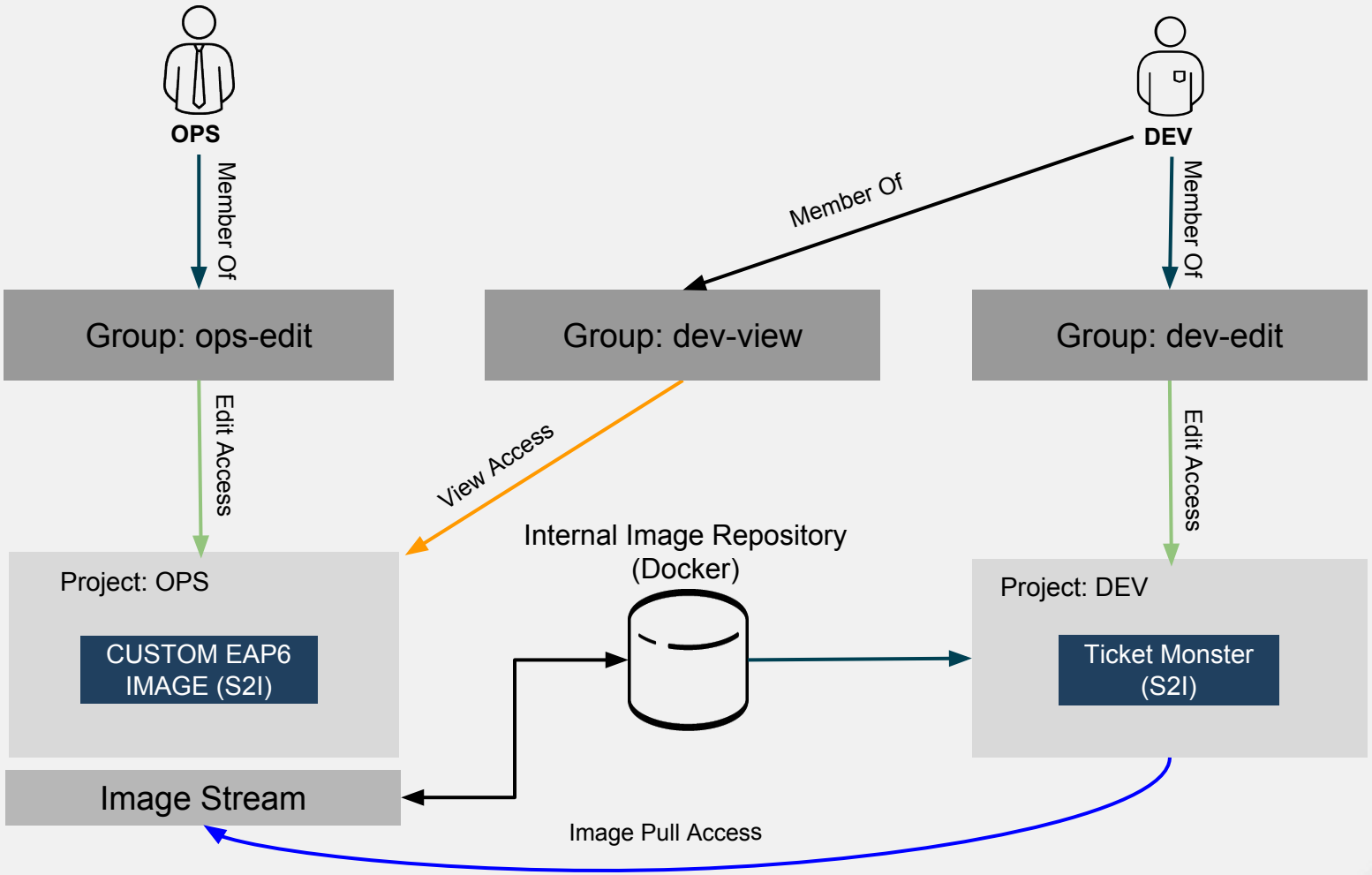
Integrated CI/CD Workflow



OSE Overview: Ops → Dev Process Flow



OSE Project and Image Security Access



OSE Non Production Environment (NPE)

Our OPS Model Benefits

- Approach only requires that the custom image be built once. After the image is built it can be used by multiple development projects
- Establishes version controlled environment specific standard images
- OPS deployments can be stacked to produce multiple images
- Methodology can be tweaked for both CI & CD environments
- Provides separation of duties

Demo Environment Details

Demo Environment

- OSE 3.1 Environment:
  - 1 - Load Balancer / NFS Server
  - 3 - Masters
  - 2 - Infranodes (registry and router)
  - 2 - Application Nodes

- Additional Components
  - 1 - Jenkins Server to act as artifact repository
  - 1 - Local GitLab instance to hold development and Ops repositories

Demo Environment - cont.

- Git Repo Details
  - Ticket Monster
    - References JBOSS web application reference architecture
    - Used to create a war artifact that will be used as the Jenkins artifact
    - https://github.com/jboss-developer/ticket-monster.git
  - Ops Git Repo
    - Git repo that holds configuration information (Infrastructure as Code) used to generate (in our case) a custom EAP6 image
    - https://github.com/themoosman/ops-custom-eap6.git
  - Ticket Monster s2i Image Build
    - Git repo that hold the scripts (s2i) used to consume development artifacts (war) and create a docker image. Said image is uploaded to a master image repository where it's pushed to dev, qa, prod, etc.
    - https://github.com/themoosman/ticket-monster-ose-s2i-build.git


Demo
redhat.

Summary

This presentation and demo has shown the following points:

- Past & Present trends of development & operations team roles within the CI/CD workflow
- Simplicity and fundamental ease of use of integrating operations into the CI/CD workflow utilizing OpenShift
- Flexibility in terms of workflow, security, and image build process
- Use of access control to implement business & security rules
- Deploy an existing application with inline changes to simulate operations team involvement


Questions?
&
Thank you for your time!
redhat.