Kubernetes SIG-Architecture

Deep Dive

Who Am I?

Timothy St. Clair
SIG Cluster Lifecycle co-lead
Conformance Cat Wrangler
Steering Committee Member
Senior Staff Engineer @VMware
@timothysc

What do you want to know?

- Distribution is better than centralization
- Community over product or company
- Automation over process
- **Inclusive is better than exclusive**
    - **Your feedback is solicited**
- Evolution is better than stagnation
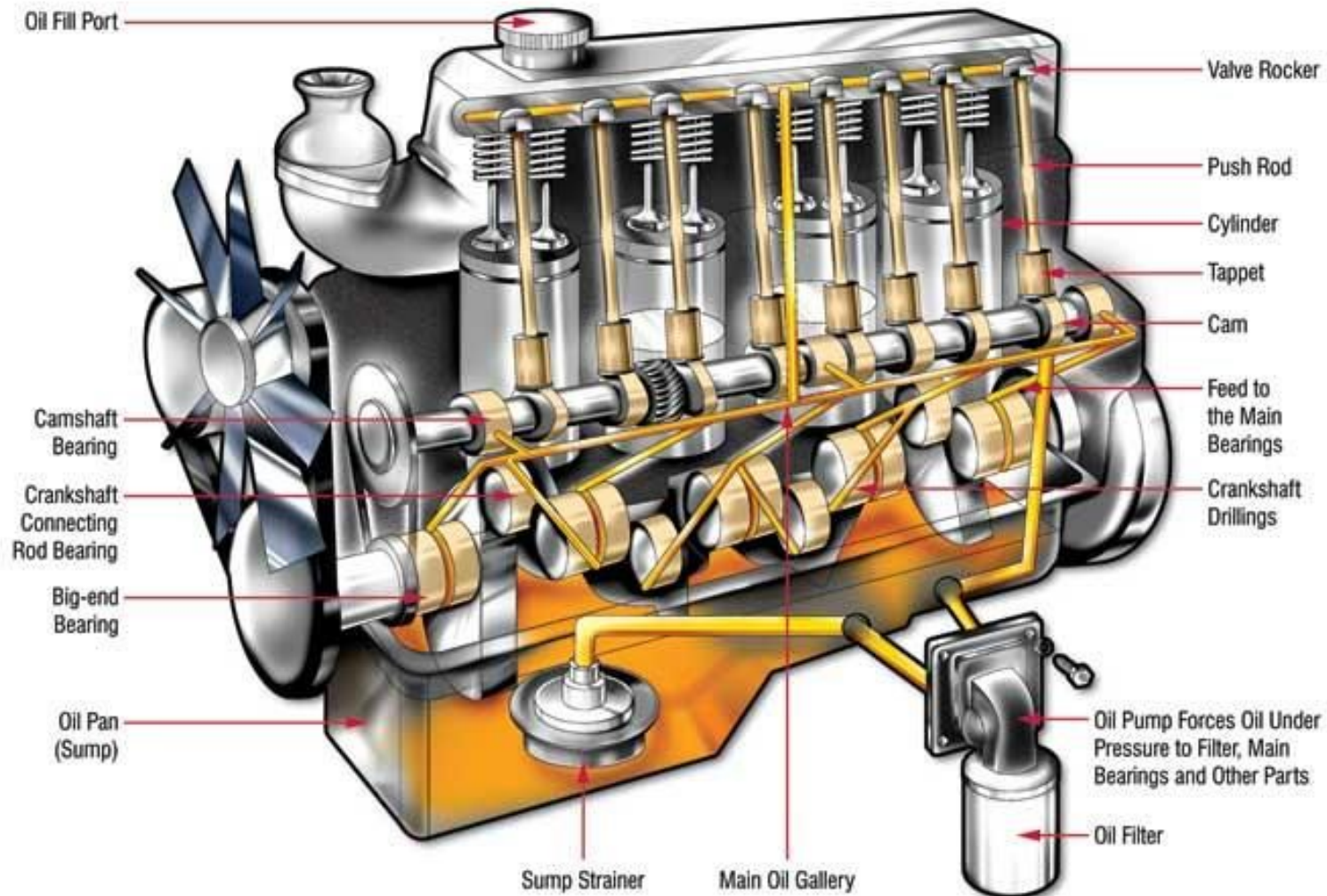
What is Kubernetes?

Demystifying Complexity

- Declarative vs Imperative
- Highly consistent KV-storage
- Proportional Controller
  - P(id)

What could go wrong?

SIG Arch Scope

The Architecture SIG maintains and evolves the design principles of Kubernetes, and provides a consistent body of expertise necessary to ensure architectural consistency over time.

- *Conformance test definitions*
- *API conventions*
- *Architectural renderings*
- *Design principles*
- *Deprecation policy*

Cross-cutting Processes

- API review process
- Conformance test review and management
- Design documentation management (KEPs)
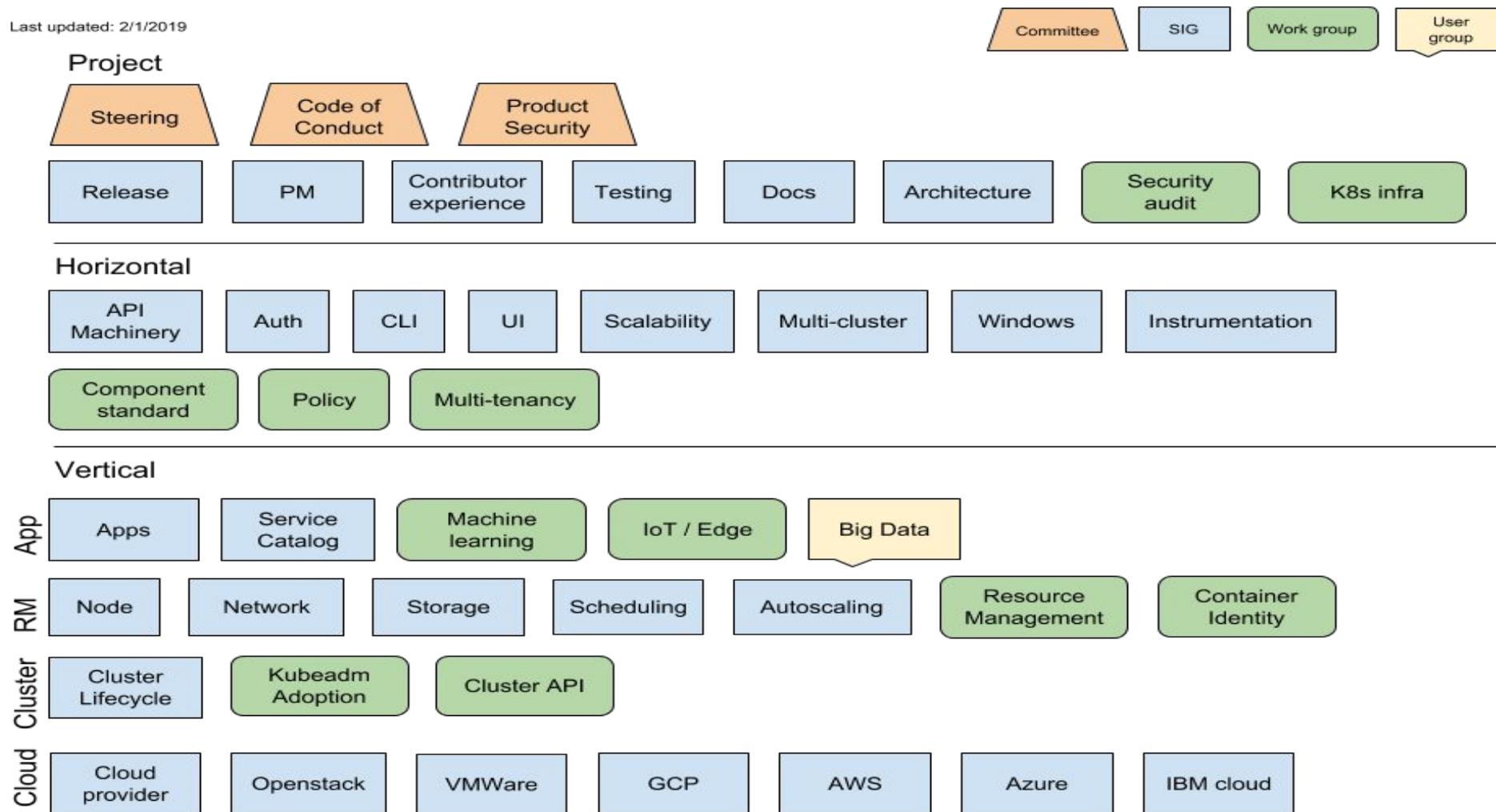- Deprecation policy management

Cross-cutting Processes

Last updated: 2/1/2019

Legend: Committee | SIG | Work group | User group

Project

- Steering
- Code of Conduct
- Product Security

- Release
- PM
- Contributor experience
- Testing
- Docs
- Architecture
- Security audit
- K8s infra

Horizontal

- API Machinery
- Auth
- CLI
- UI
- Scalability
- Multi-cluster
- Windows
- Instrumentation

- Component standard
- Policy
- Multi-tenancy

Vertical

App
- Apps
- Service Catalog
- Machine learning
- IoT / Edge
- Big Data

RM
- Node
- Network
- Storage
- Scheduling
- Autoscaling
- Resource Management
- Container Identity

Cluster
- Cluster Lifecycle
- Kubeadm Adoption
- Cluster API

Cloud
- Cloud provider
- Openstack
- VMWare
- GCP
- AWS
- Azure
- IBM cloud

What other kinds of issues?

- Ambiguous behavioral questions
  - Inconsistencies in behavior across resources
- Unanswered questions
- Anything where TL/Chairs/Owners conflict
- Come with [KEPs](#) and details!

Sub Projects

- Architecture and API
  - Document design principles
  - Document and evolve system architecture
  - Reviewing, Curating extension patterns
- Code Organization
  - Repository structure, branching, vendoring
- Conformance Definition
  - Review, approve changes to conformance test suite

API Review

- https://github.com/kubernetes/community/blob/master/sig-architecture/api-review-process.md
- https://github.com/orgs/kubernetes/projects/13

Conformance Test & Promotion

- https://github.com/kubernetes/community/blob/master/contributors/devel/sig-architecture/conformance-tests.md#conformance-test-requirements
- https://github.com/orgs/kubernetes/projects/9

Code Organization

- https://github.com/orgs/kubernetes/projects/27

Where are we going?

- Locking down the core

- APIs as CRDs

- Focus on extensions

  ○ Let the ecosystem grow and distill the best patterns

- Buildout conformance

- Cleanup / Restructure + Reliability (.0)

- Organization Scaling

- KEPs KEPs KEPs

How you can participate?

- Attend the main and sub project meetings
- Follow along on project boards and mailing list discussions and #sig-architecture
- Find something of interest you can help with
- Speak up - offer your thoughts and ideas, ask questions for background/history etc
- Help with Issue Triage, PR reviews