

Policy-as-code with Kubewarden



Divya Mohan

- Senior Technical Evangelist at SUSE
- Documentation maintainer @ Kubernetes
- CNCF Ambassador
- KCNA exam co-creator



Agenda

1. What is Kubewarden?
2. Architecture
3. PSP, PSA, and Kubewarden
4. Demo # 1
5. What's new in Kubewarden v1.3
6. Demo # 2



Supporting Material

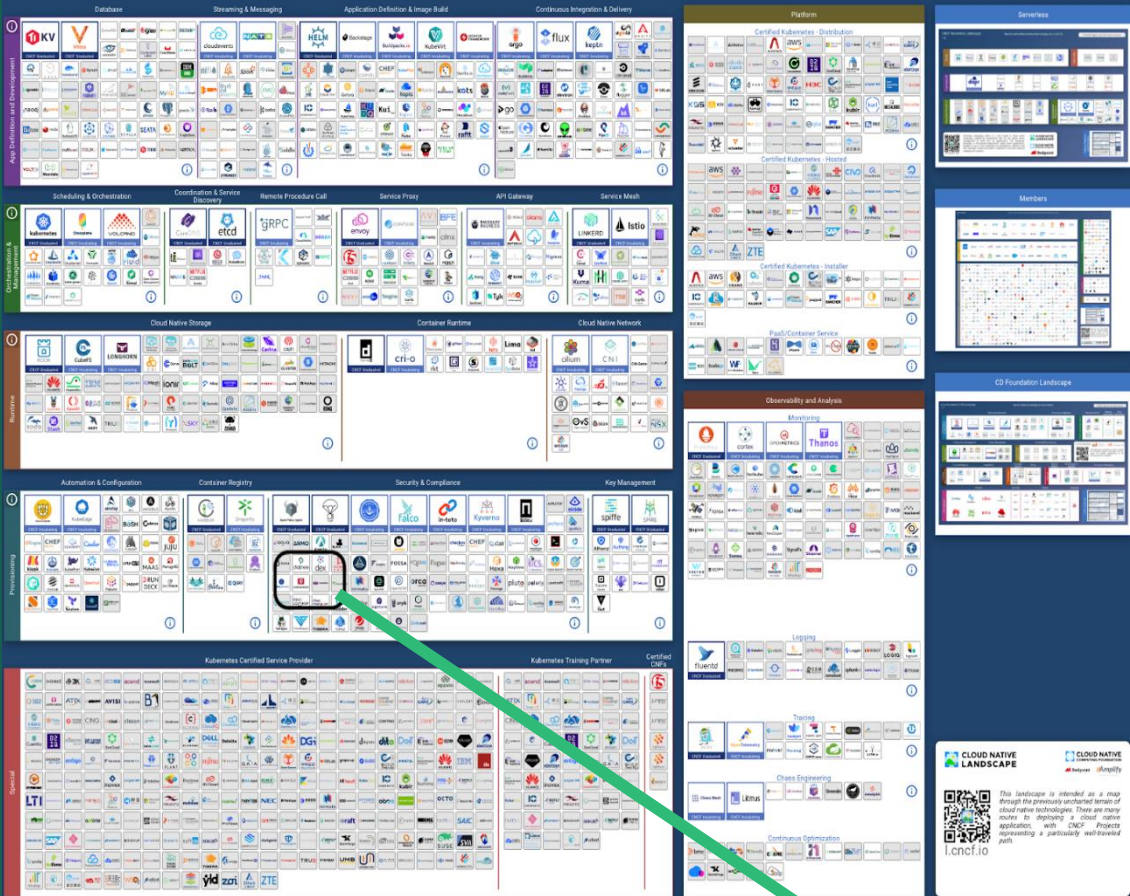
manifests-2	Create resource1.yaml
manifests	Create user-psp-policy.yaml
LICENSE	Initial commit
README.md	Update README.md
README.md	
<h2>Policy-as-code with Kubewarden</h2>	
<h3>Topics covered</h3>	
<ul style="list-style-type: none">• Introduction• Architecture• PSP, PSA, and Kubewarden• Demo 1• What's new in Kubewarden v1.3• Demo 2	



What is Kubewarden?



Policy engine for Kubernetes to
simplify policy-as-code



CNCF Sandbox project



What's the secret sauce?



Dynamic admission control



What's the secret sauce?

- Users can write Kubernetes policies in their favorite programming language
 - Caveat: The language can compile to Wasm binaries
- Currently supports:
 - Rust
 - Go
 - Swift

What's the secret sauce?

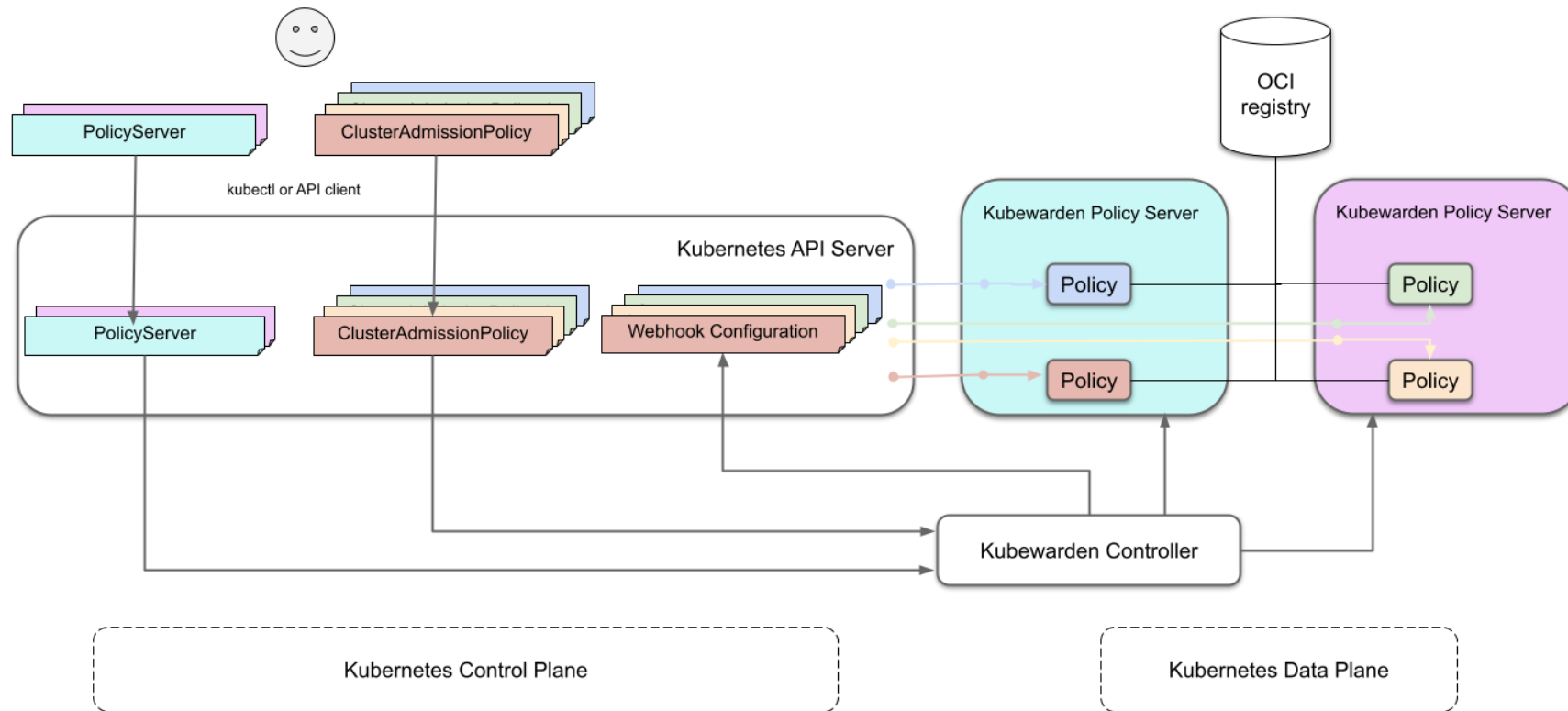
- You can also reuse (almost) all your existing **Rego** policies!
 - Some built-in functions are SDK-dependent i.e. Kubewarden has to implement them
 - Built-ins required by the majority of K8s users are supported.
- Distribution channels:
 - Served by a web server
 - Published & stored inside an OCI compliant registry as ICU artifacts



Architecture



Architecture



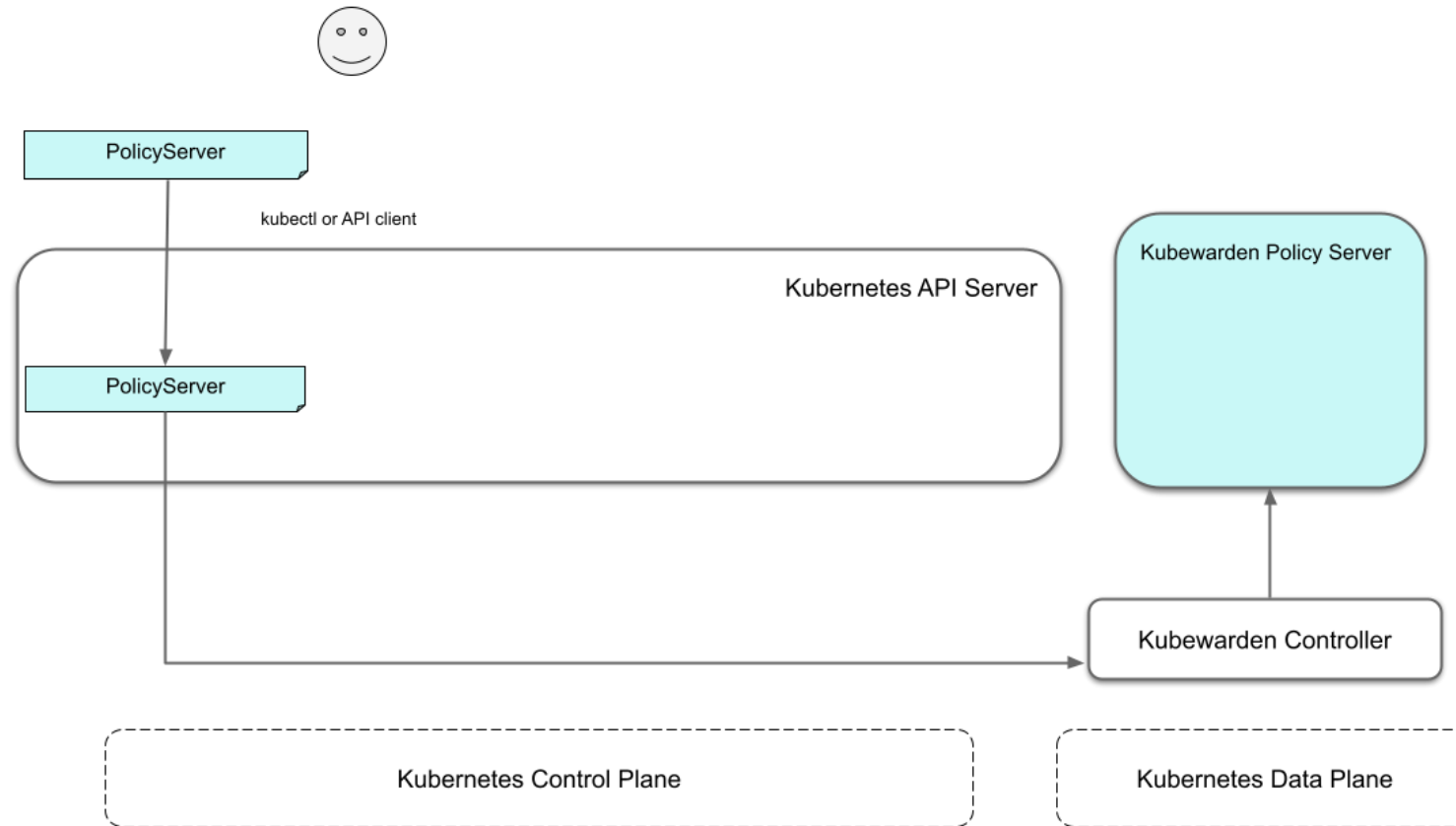
Courtesy: <https://docs.kubewarden.io>



Request flow



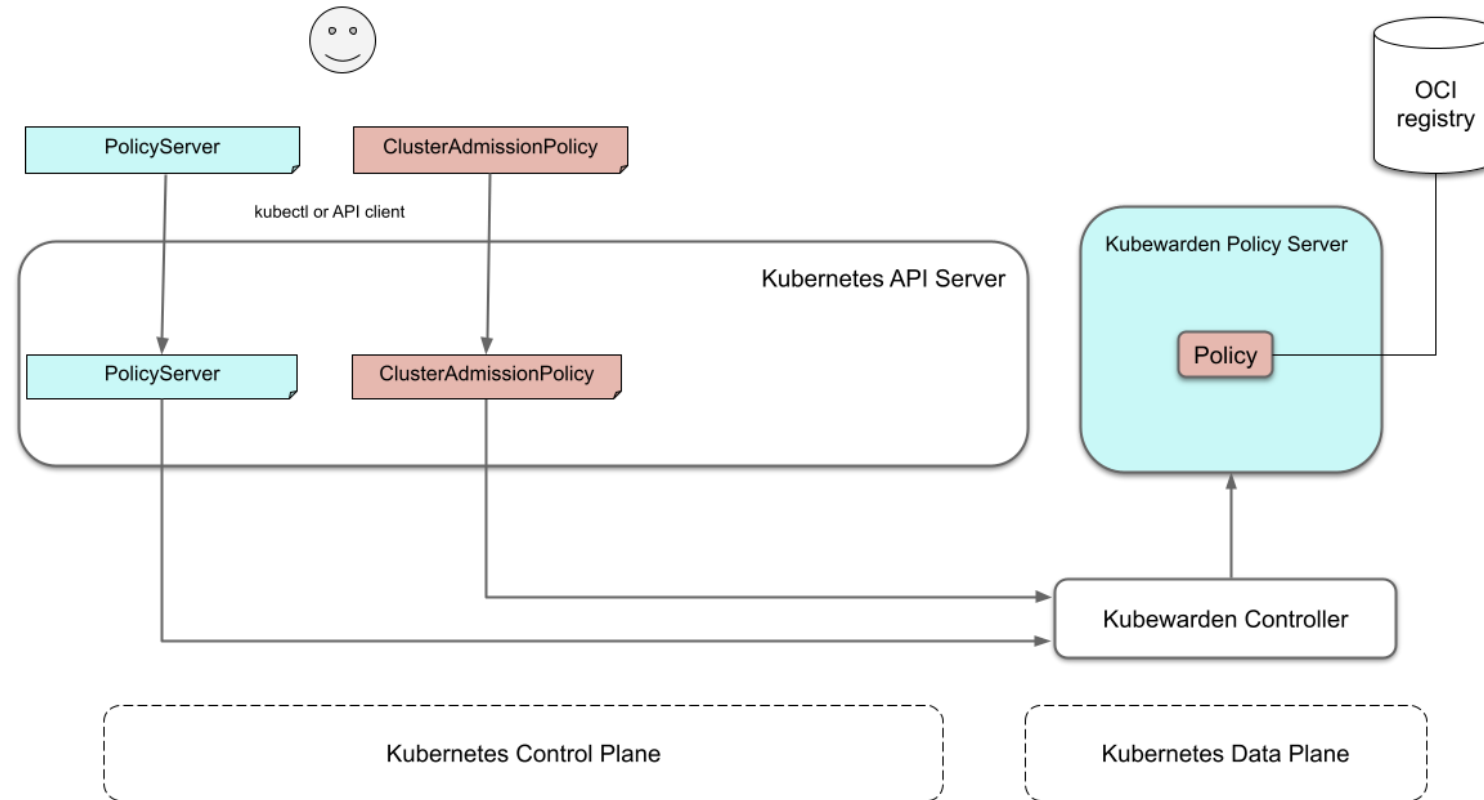
Default policy server



Courtesy: <https://docs.kubewarden.io>



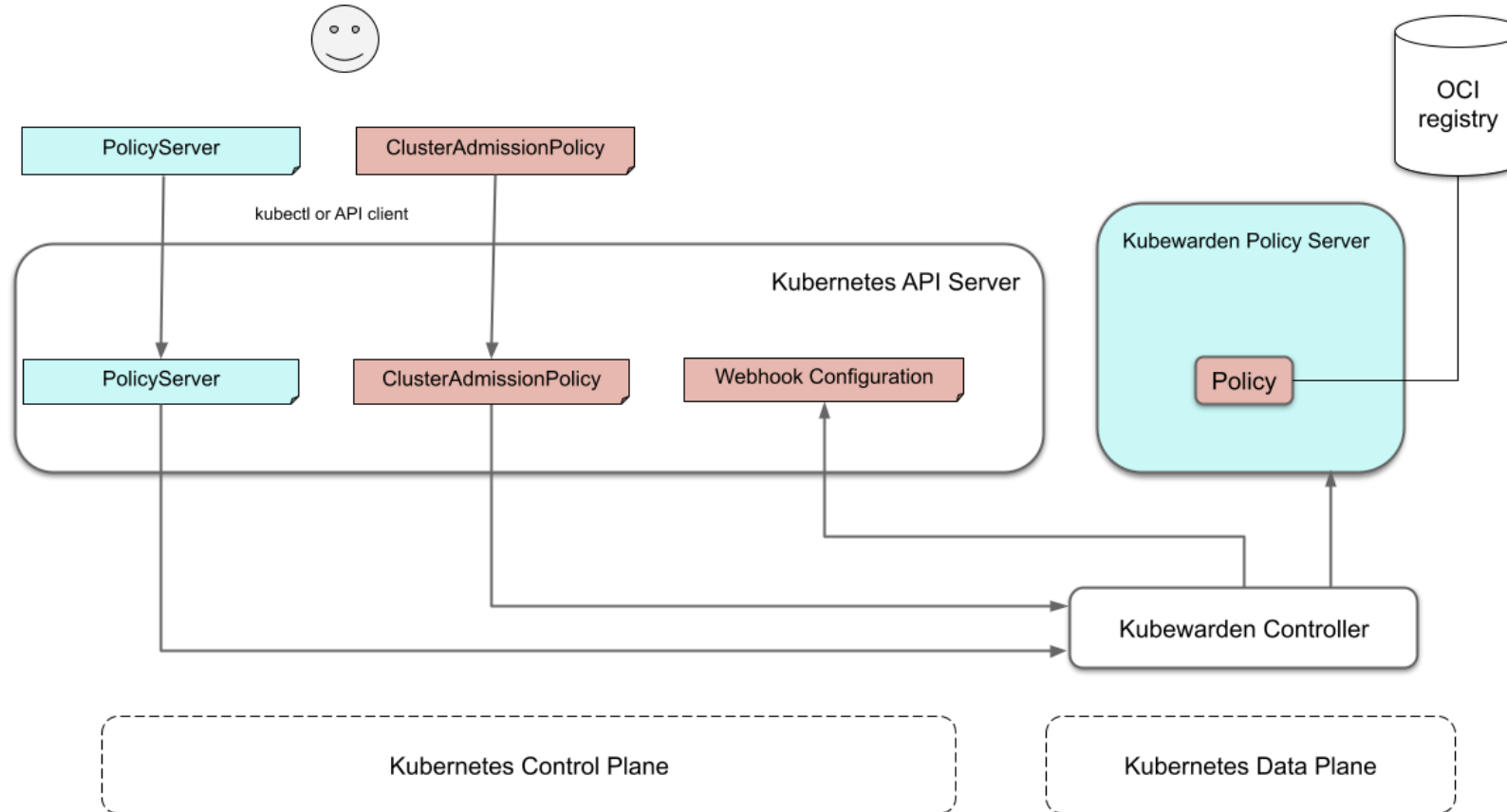
Reconciliation of policy server



Courtesy: <https://docs.kubewarden.io>



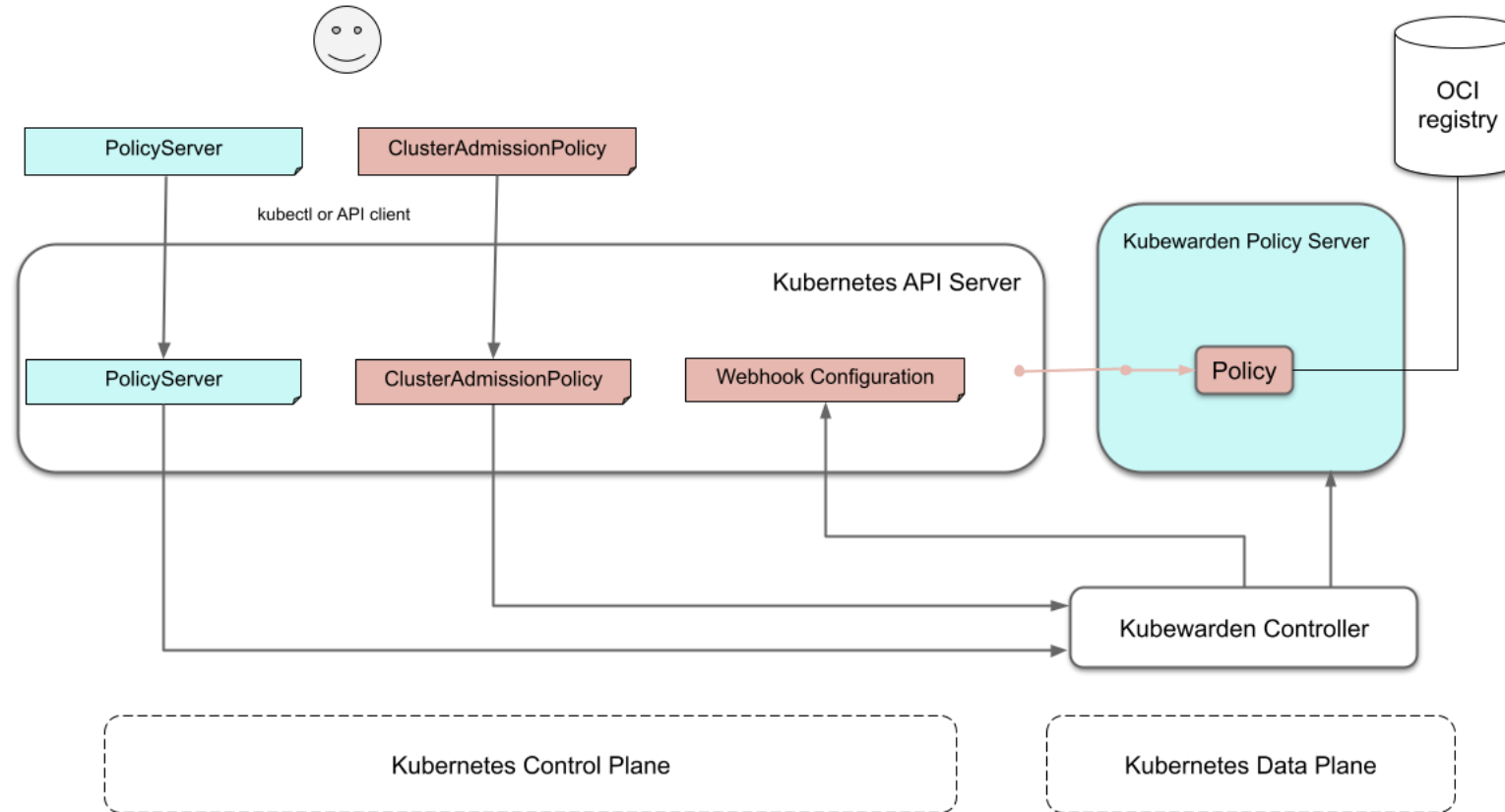
Making Kubernetes aware of the policy server



Courtesy: <https://docs.kubewarden.io>



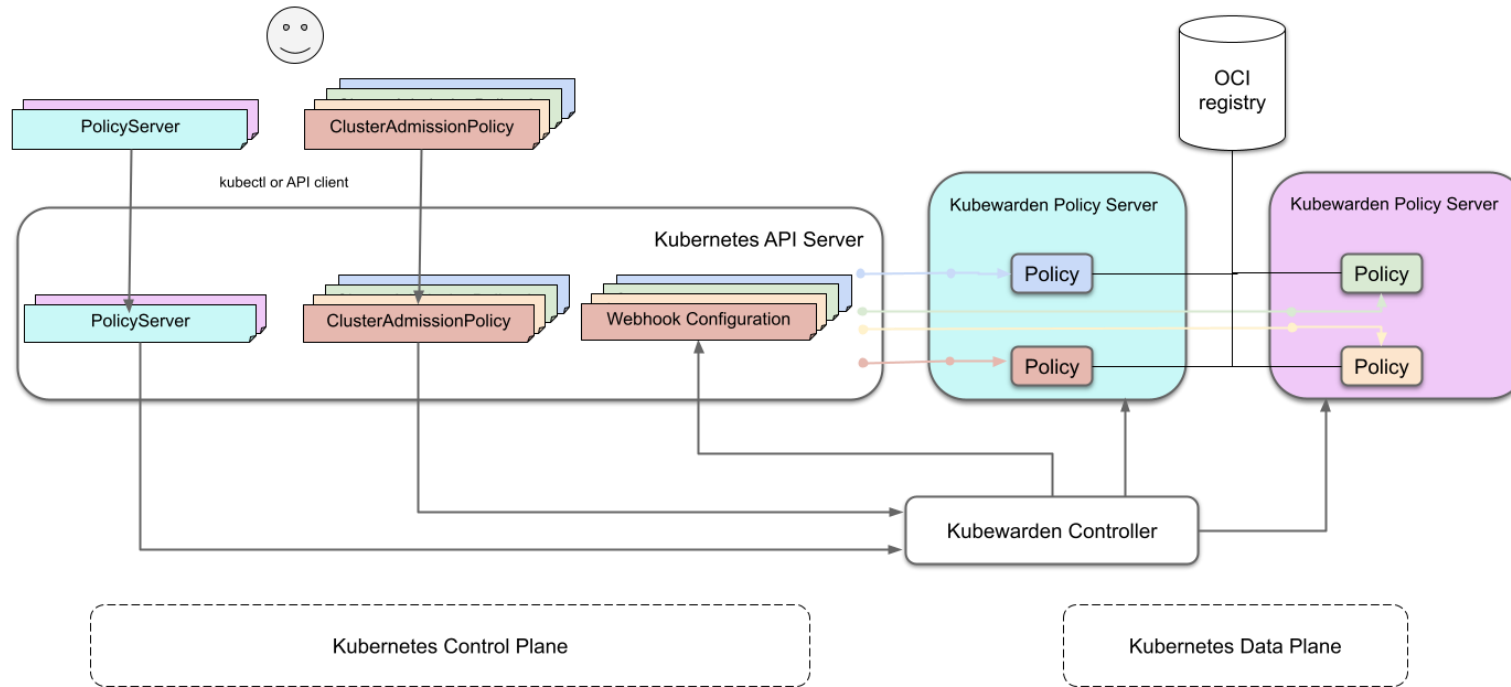
Policy in action



Courtesy: <https://docs.kubewarden.io>



Handling multiple policies



Courtesy: <https://docs.kubewarden.io>



PSP, PSA, and Kubewarden



PodSecurityPolicy

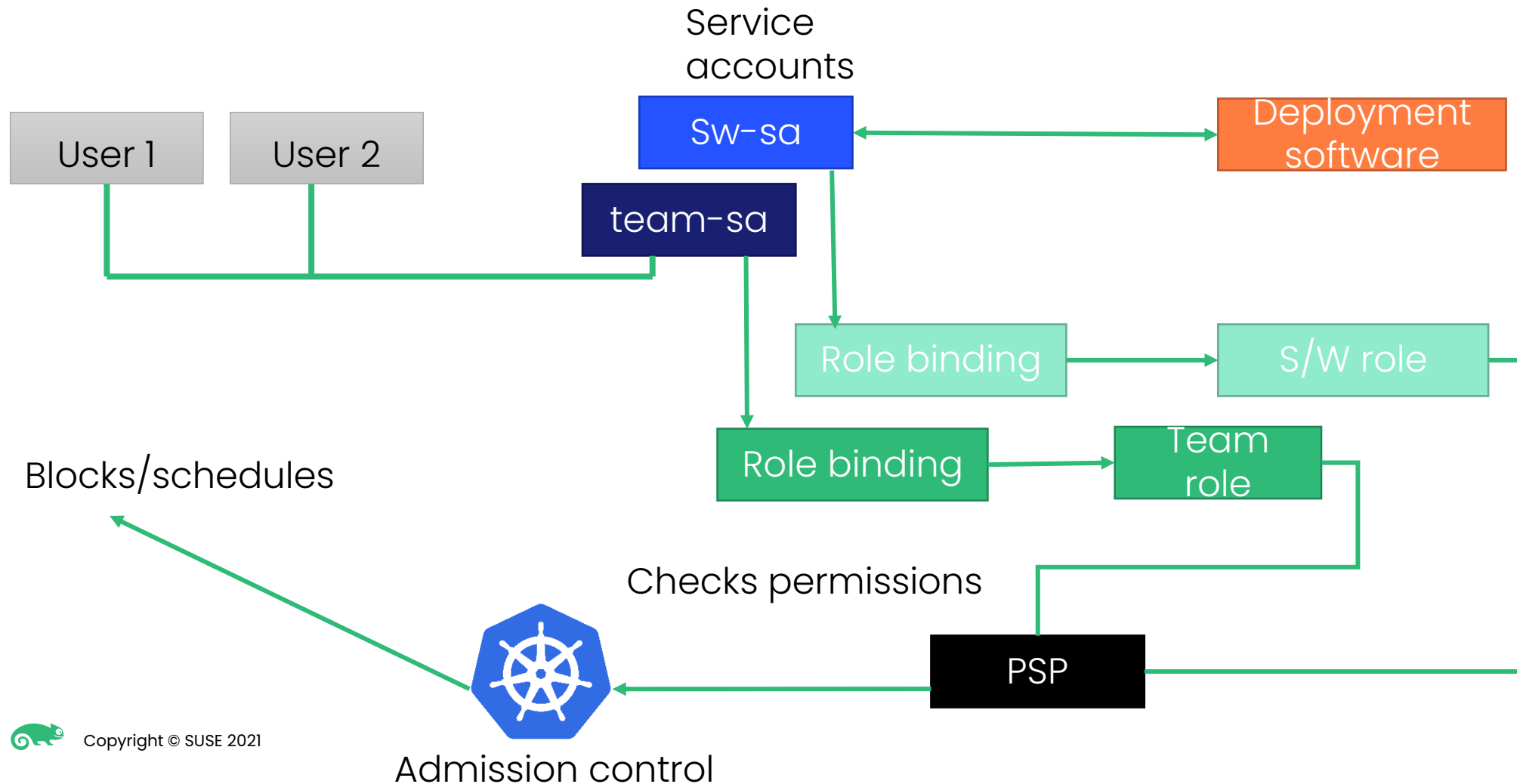
- Framework to ensure that Pods run only with appropriate privileges & can only access appropriate resources
- Kubernetes RBAC links PSPs to users/services through the roles they have
- Enforce the concept of least privilege



kubernetes



PSP in action



What was the problem with PSP?

- Easy to grant broader permissions than intended
- Difficult to determine which PSP applies in a given situation



kubernetes



Pod Security Admission

- Different isolation levels for Pods based on Pod Security Standards
- Applied at the namespace level
- Allows you to define the behavior of pods in a clear & consistent fashion.



kubernetes



What's the problem with Pod Security Admission?

- As of Kubernetes v1.25
 - No mutation capabilities
 - Higher level objects are evaluated ONLY when audit/warn modes are enabled.



kubernetes



Kubewarden!

- Can be used to replace PSP
- Intended to complement Pod Security Admission
- Integrate Kubewarden into a Pod Security Admission profile to mitigate limitations



Demo #1



What's new in Kubewarden v1.3?



What's new in Kubewarden v1.3?

- Joined the CLO Monitor Initiative
 - Currently A-rated with a score of 97%
- Reduced startup time for Policy server
- Ability to handle Sigstore signatures produced using a PKCS11 token



What's new in Kubewarden v1.3?

- New policies that are backward compatible!
 - Environment Variable Scanner Policy
 - Environment Variable Compliance Policy
 - volumeMounts policy
 - deprecated-api-versions policy
- Expansion of scope for existing policies



Demo #2



Resources

- Kubernetes website
 - <https://kubernetes.io/docs/reference/access-authn-authz/extensible-admission-controllers/>
 - <https://kubernetes.io/docs/concepts/security/pod-security-policy/>
 - <https://kubernetes.io/docs/concepts/security/pod-security-admission/>
 - <https://kubernetes.io/docs/concepts/security/pod-security-standards/>
- Kubewarden website
 - <https://kubewarden.io>
- [Artifact Hub](#)



Where can you find us?

- [Kubernetes slack](#) (#kubewarden channel)
- [Twitter](#)
- [YouTube channel](#)





Q&A



Thank You

© 2020 SUSE LLC. All Rights Reserved. SUSE and the SUSE logo are registered trademarks of SUSE LLC in the United States and other countries. All third-party trademarks are the property of their respective owners.