Informational

for workload identity

Former “Best Current Practices for Workload Identity”

Benedikt Hofmann

Hannes Tschofenig

Edoardo Giordano

Yaroslav Rosomakho

Arndt Schwenkschuster

Since IETF 120

- Agreement to move to “informational” rather than “BCP”
- Document current practices without strong recommendations

Various patterns in the industry

kubernetes

Old scope

spiffe

Added

Other workload platforms?!

Input wanted

Google Cloud Platform

aws

Azure

Added

Jenkins

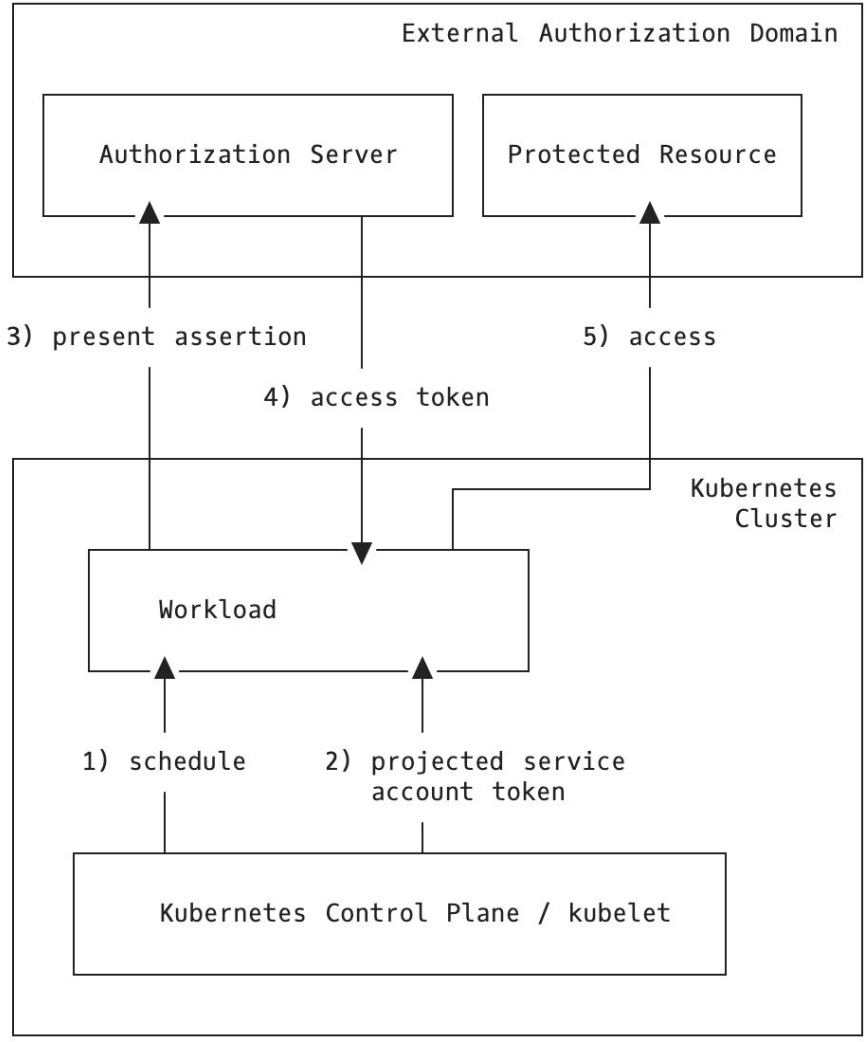
GitHub

GitLab

Added

Kubernetes

1) Workload (aka “Pod”) gets scheduled by the Kubernetes Control Plane and kubelet
2) Kubelet requests service account token (PSAT) on behalf of the workload from the control plane and projects it into the workloads filesystem.
3) Workload presents PSAT as an assertion to an AS in an external domain.
4) AS returns an access token.
5) Workload access the protected resource.



Cloud providers

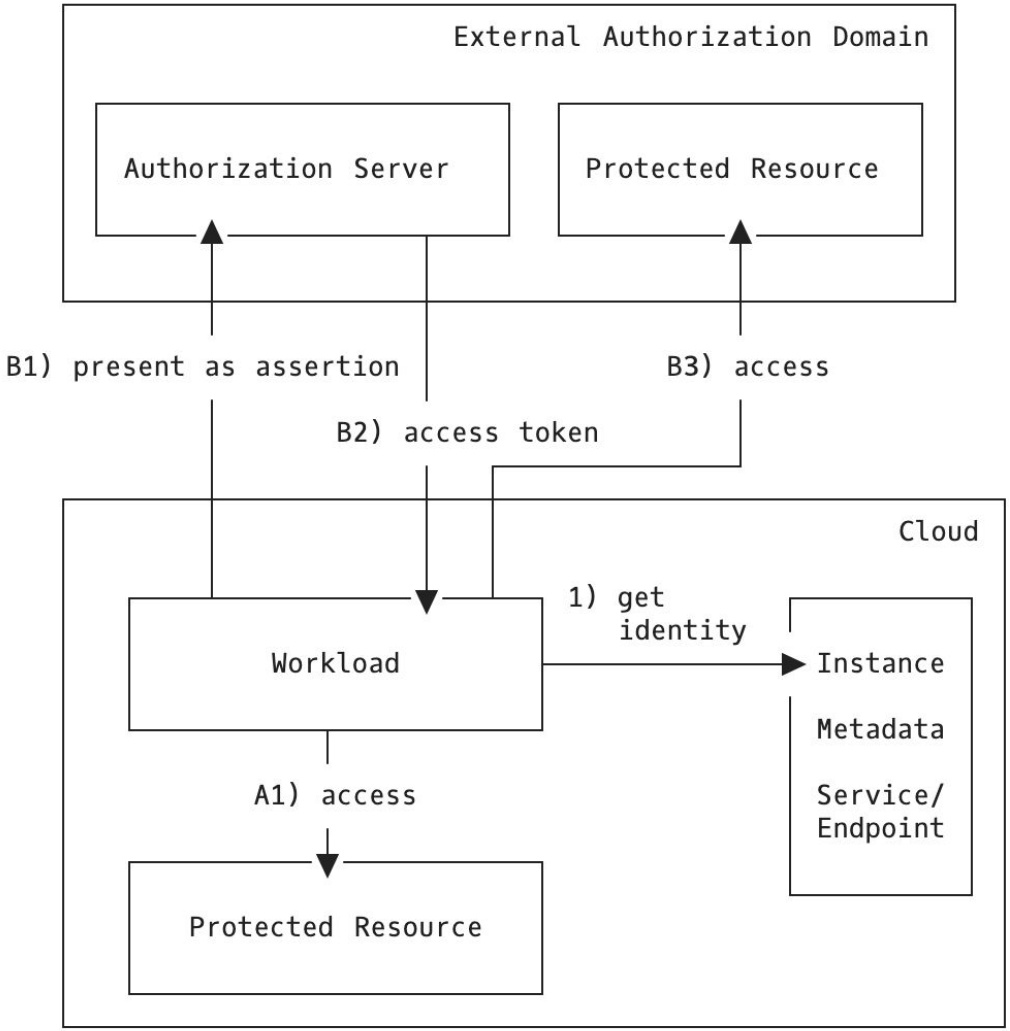
1) Workload that runs as VM, Serverless, low/no-code, etc. requests a credential from the “Instance Metadata Service”.

—- SAME DOMAIN —

A1) Workload uses the token directly to access resources in the same domain.

— OTHER DOMAIN —

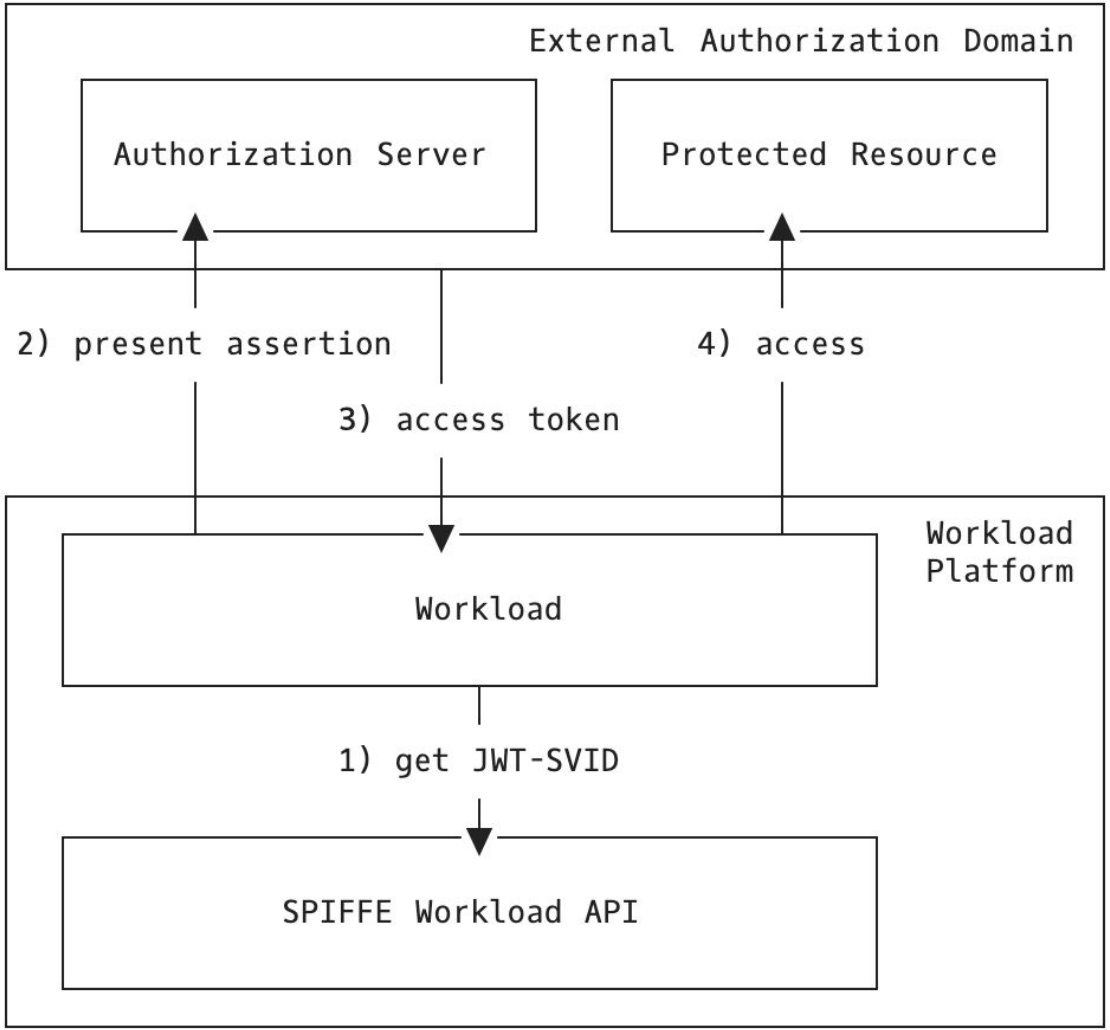
B1) Workload presents token as an assertion to external AS
B2&3) Workload gets access token and uses it.



SPIFFE

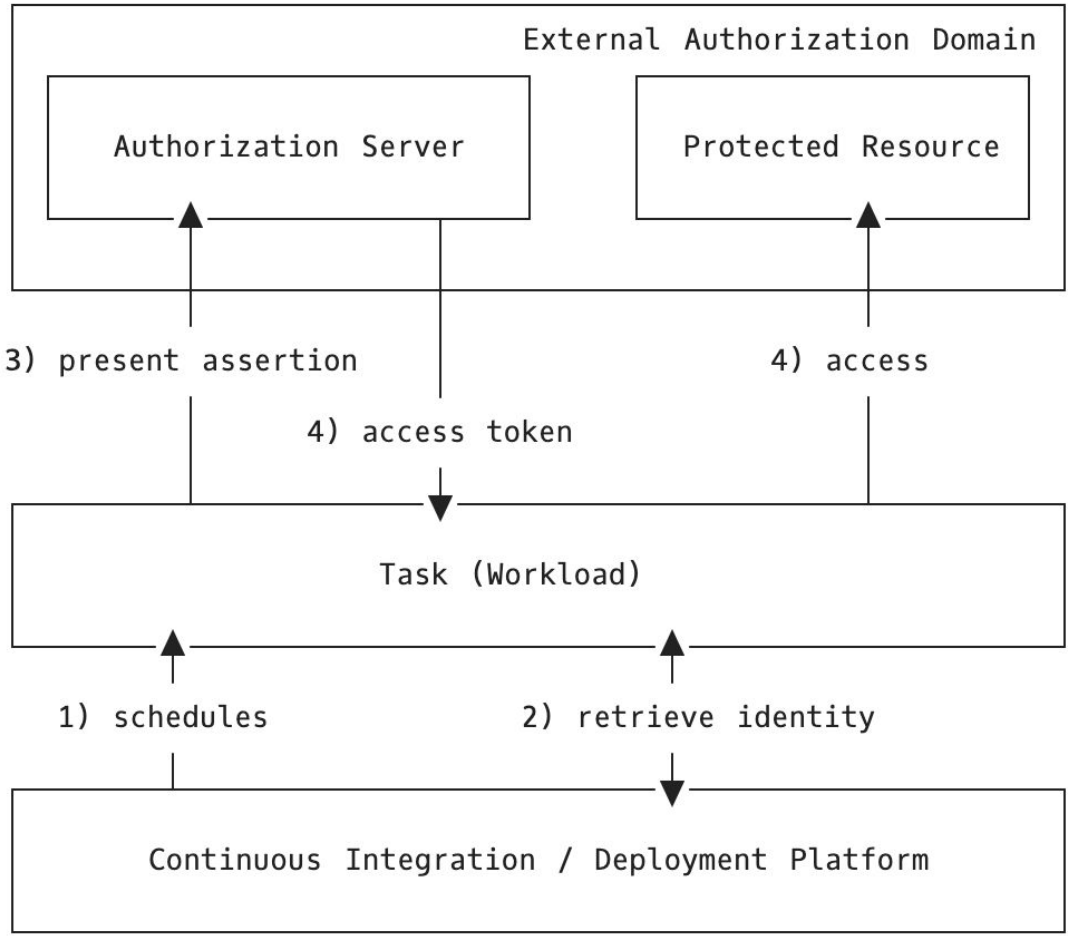
1) Workload uses the SPIFFE Workload API to get a JWT-SVID.
2) The Workload uses the JWT-SVID as an assertion to the external AS
3) External AS returns access token
4) Workload accesses protected resource.

Workload platform here can be anything that SPIFFE supports, from VM to Kubernetes, to serverless.



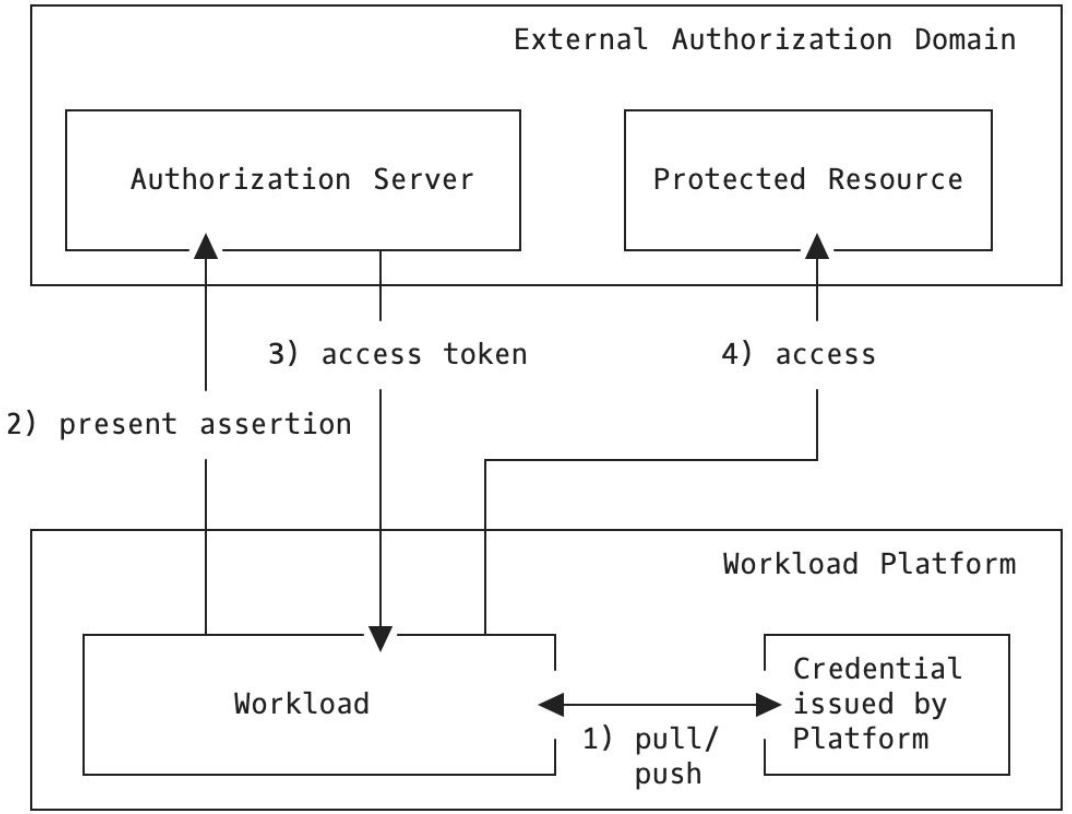
CI/CD

1) CI/CD platform schedules a task/job/workflow
2) Workload retrieves identity from the platform (or got provisioning with it)
3) Workload uses identity credential as an assertion towards an external AS
4) AS returns access token
5) Workload access protected resource.



Common pattern

1) Workload receives a credential issued by the platform. (Most of the time a JWT token).
2) Workload presents this credential as an assertion to a AS in an external domain.
3) AS returns an access token.
4) Workload access the protected resource.



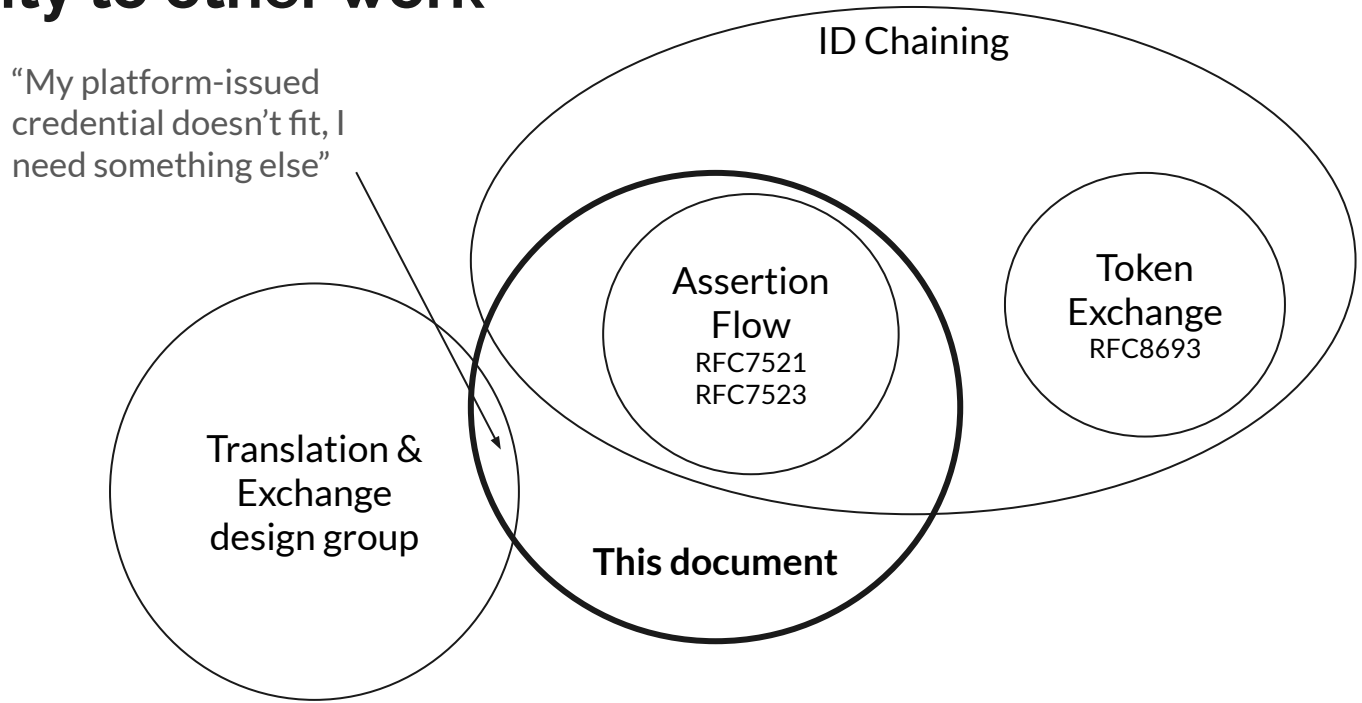
External Authorization Servers

“How to change a platform-issued token into an access token of the other domain”

Security considerations

- Issuer, subject and audience validation - or “Based on what claims do I accept assertions?”.
- Token types
- Custom claims are important for context
- Token lifetime
- Workload lifecycle and invalidation
- Proof of possession
- More ..

Similarity to other work



Discussion: New direction (4 remaining)

Is the WG happy with the “new” direction of this document?

Discussion: Document patterns on Authorization Servers (3 remaining)

- Should this document:
  - Outline patterns on Authorization Server (Assertion flow vs Token Exchange vs custom)?
  - Have an opinion/give recommendations which flow authorization servers should pick?

Discussion: More platforms/patterns (2 remaining)

Does the WG see more platforms or patterns this document should outline and consider?

As of now we have:

- Kubernetes
- SPIFFE
- CI/CD
- Cloud providers

Discussion: Delivery mechanisms (1 remaining)

Issue #7

There's many ways to delivery credentials to the workload, the most common are:

  - Environment variables
  - Local API (socket or web)
  - File system
- Does the WG think, it is worth outlining these mechanisms in this document?
- Any other pattern?

Discussion: Name (last)

“Best Current Practice for Workload Identity” doesn’t fly anymore as we are not a BCP.

At the moment it’s “OAuth 2.0 Client Assertion in Workload Environments”

Ideas? Opinions?