Universidade Federal
de Campina Grande

COMPUTAÇÃO UFCG

Strengthening the SPIRE id provisioning workflow

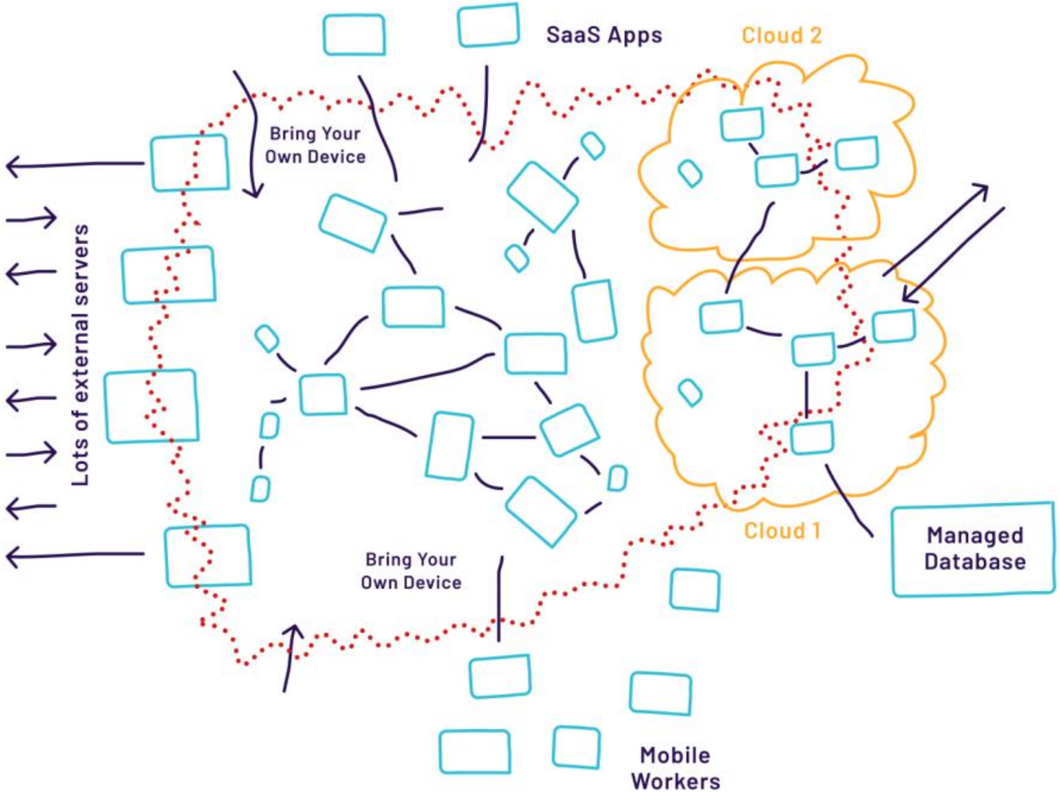
andrey@computacao.ufcg.edu.br

82nd IFIP WG 10.4 Meeting
June 26th, 2022

LABORATÓRIO
DE SISTEMAS
DISTRIBUÍDOS

Zero Trust - Motivation

Permeable perimeters



Solving the Bottom Turtle (2020, link)

Zero Trust - Principles

Identity

- All data sources and computing elements are resources

Authorization

- Communication is secured regardless of location
- Access is granted on a per-session basis
- Authentication and authorization are enforced

Monitoring

- Relevant assets are monitored
- Access is determined by a dynamic policy
- Monitoring is used to improve the security posture

NIST SP 800-207 Zero Trust Architecture

Robust Identity Provisioning

Root of the Zero Trust approach

Goals:

- Continuously evaluate workloads and infrastructure
- Automatically issue short-term identities
- Identities bound to software/environment (not to other identities or secrets)
- Have “adequate” verification mechanisms

Challenges: automated, simple to bootstrap, compatible with the threat model

The SPIFFE Standard (CNCF)

SPIFFE ID

spiffe://example.org/app1/client

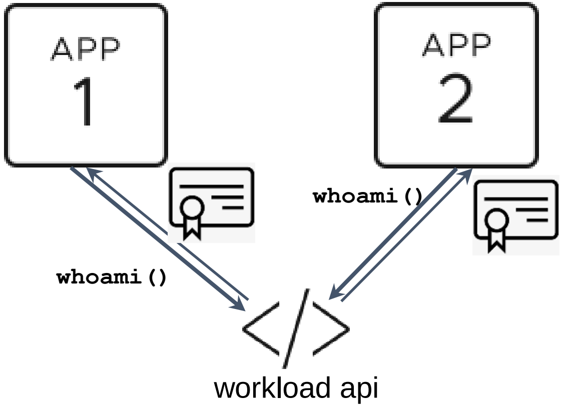
Domain + Workload Id

Id is opaque or human-friendly

SPIFFE Verifiable Identity Document (SVID)

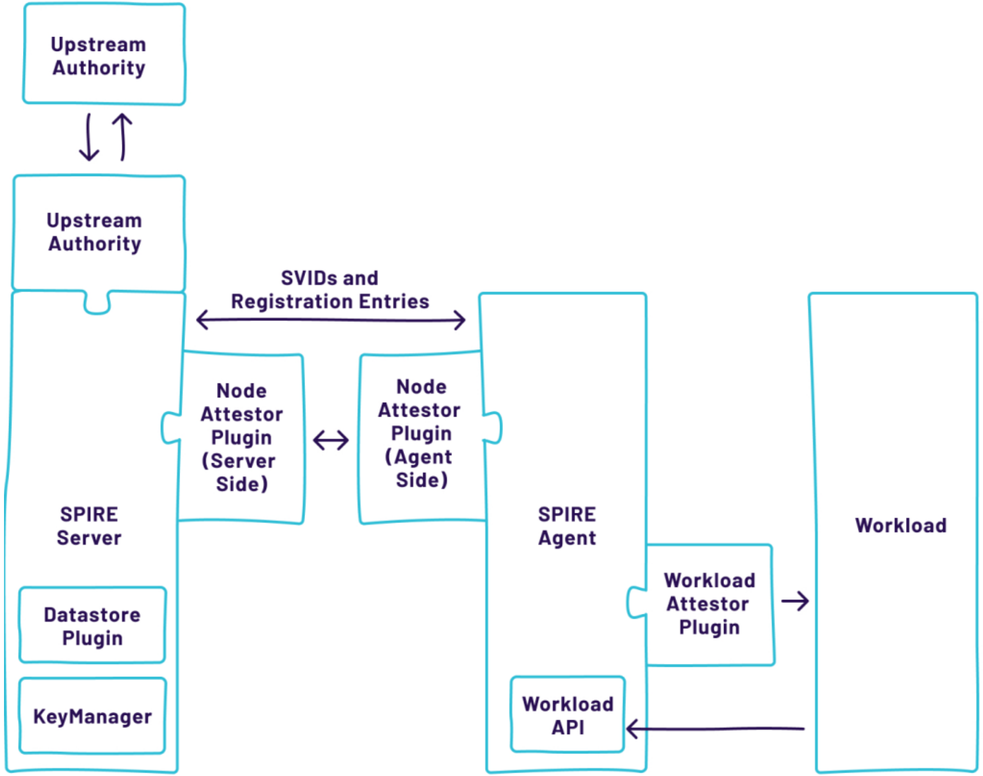
X.509 with ID in URI SAN

SPIFFE Workload API



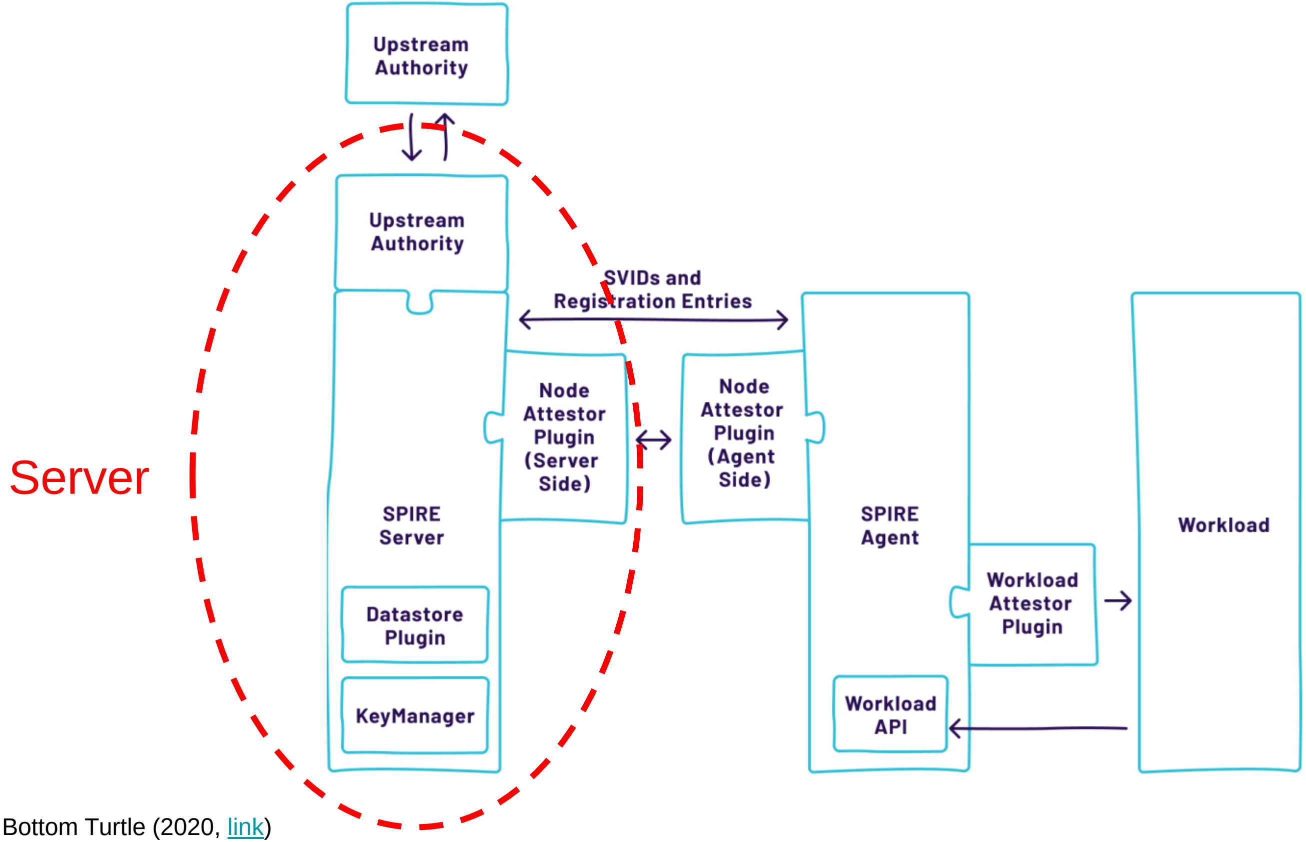
Locally cached certs and bundles

Reference Implementation - SPIRE (CNCF)



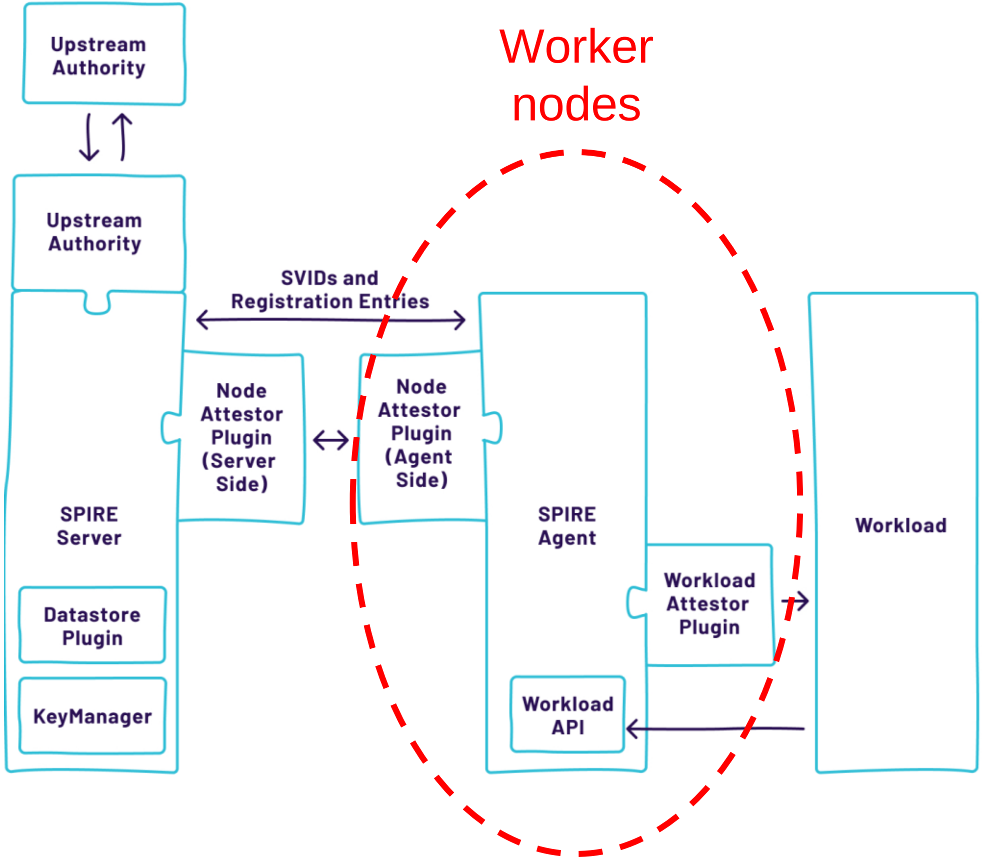
Solving the Bottom Turtle (2020, link)

Reference Implementation - SPIRE (CNCF)



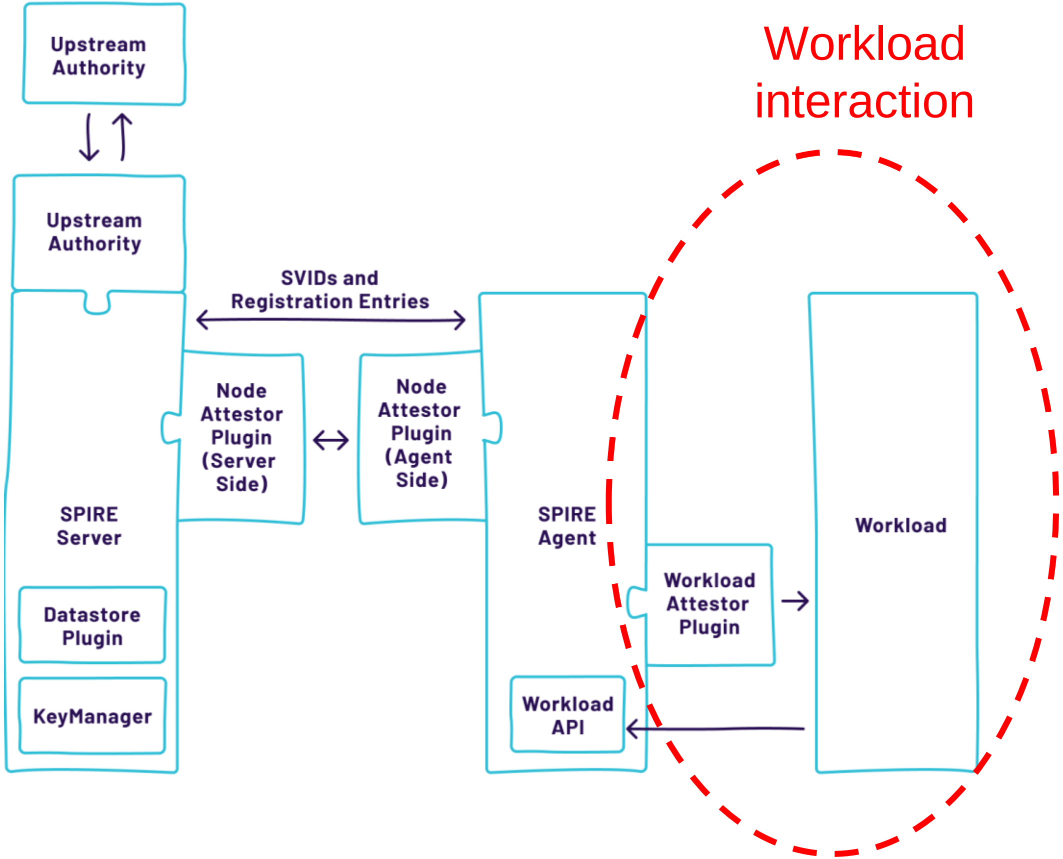
Solving the Bottom Turtle (2020, link)

Reference Implementation - SPIRE (CNCF)



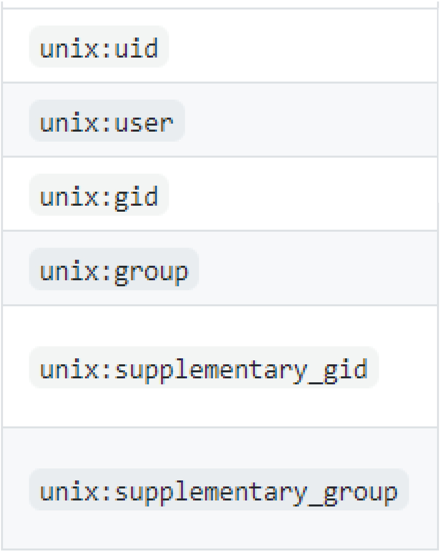
Solving the Bottom Turtle (2020, link)

Reference Implementation - SPIRE (CNCF)

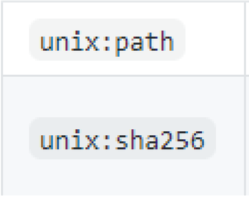


Solving the Bottom Turtle (2020, link)

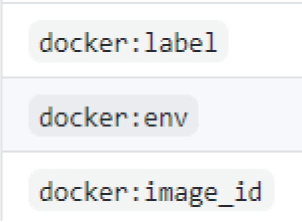
Properties for Workload Attestation (Examples)



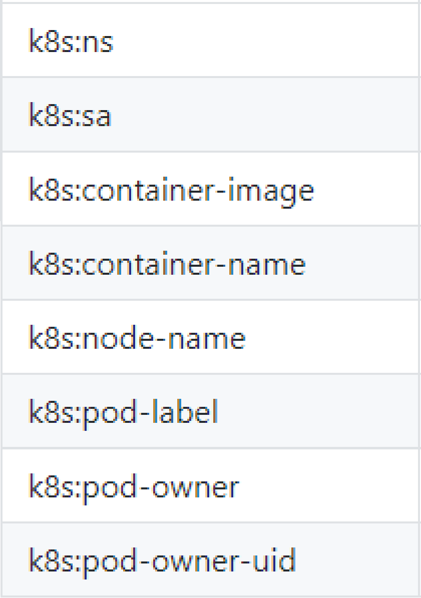
unix:uid
unix:user
unix:gid
unix:group
unix:supplementary_gid
unix:supplementary_group



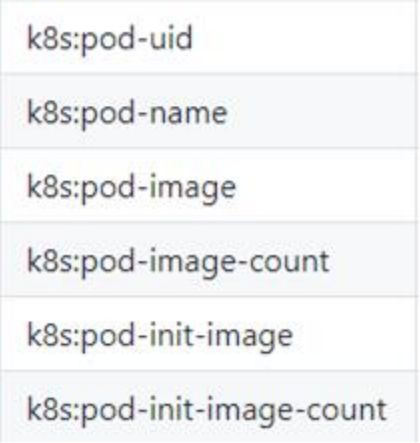
unix:path
unix:sha256



docker:label
docker:env
docker:image_id



k8s:ns
k8s:sa
k8s:container-image
k8s:container-name
k8s:node-name
k8s:pod-label
k8s:pod-owner
k8s:pod-owner-uid



k8s:pod-uid
k8s:pod-name
k8s:pod-image
k8s:pod-image-count
k8s:pod-init-image
k8s:pod-init-image-count

Registering workloads with SPIFFE IDs in the SPIRE Server

Threat model and research demands

Currently (by the community):

- Code audits and security evaluation done
- Evaluated different attacker capabilities, but no focus on internal attacks

Demands (expand attacker capabilities):

- Support new types of plugins considering TEEs
- Protecting agent and server processing and temporary space with TEEs

Opportunities:

- Protecting storage against Sybil and rollback
- Trusted time
- Fulfilling Zero Trust goals: richer dynamic/behavior verification

Thank you!

andrey@computacao.ufcg.edu.br

Acknowledgements:
ZTPO Project (UFCG, CEEI/EMBRAPII & HPE)
SPIFFE/SPIRE Community
TU Dresden

LABORATÓRIO DE SISTEMAS DISTRIBUÍDOS