Authentication in a Software-Defined World with SPIFFE on Industrial Edge

Sören Stelzer
Principal Engineer - Cloud/Edge Computing
& Co-Lead Architect Industrial Edge

soeren.stelzer@siemens.com

Dr. Andreas Reiter
Senior Key Expert – Cybersecurity

andreasreiter@siemens.com

Agenda

What problem are we trying to solve?

- Brief Introduction to Industrial Edge Platform/Ecosystem
- Challenges & Goals

Introduction to SPIFFE and SPIRE

- Terminologies
- Typical scenarios
- Identity tokens

How the Industrial Edge platform has increased trust

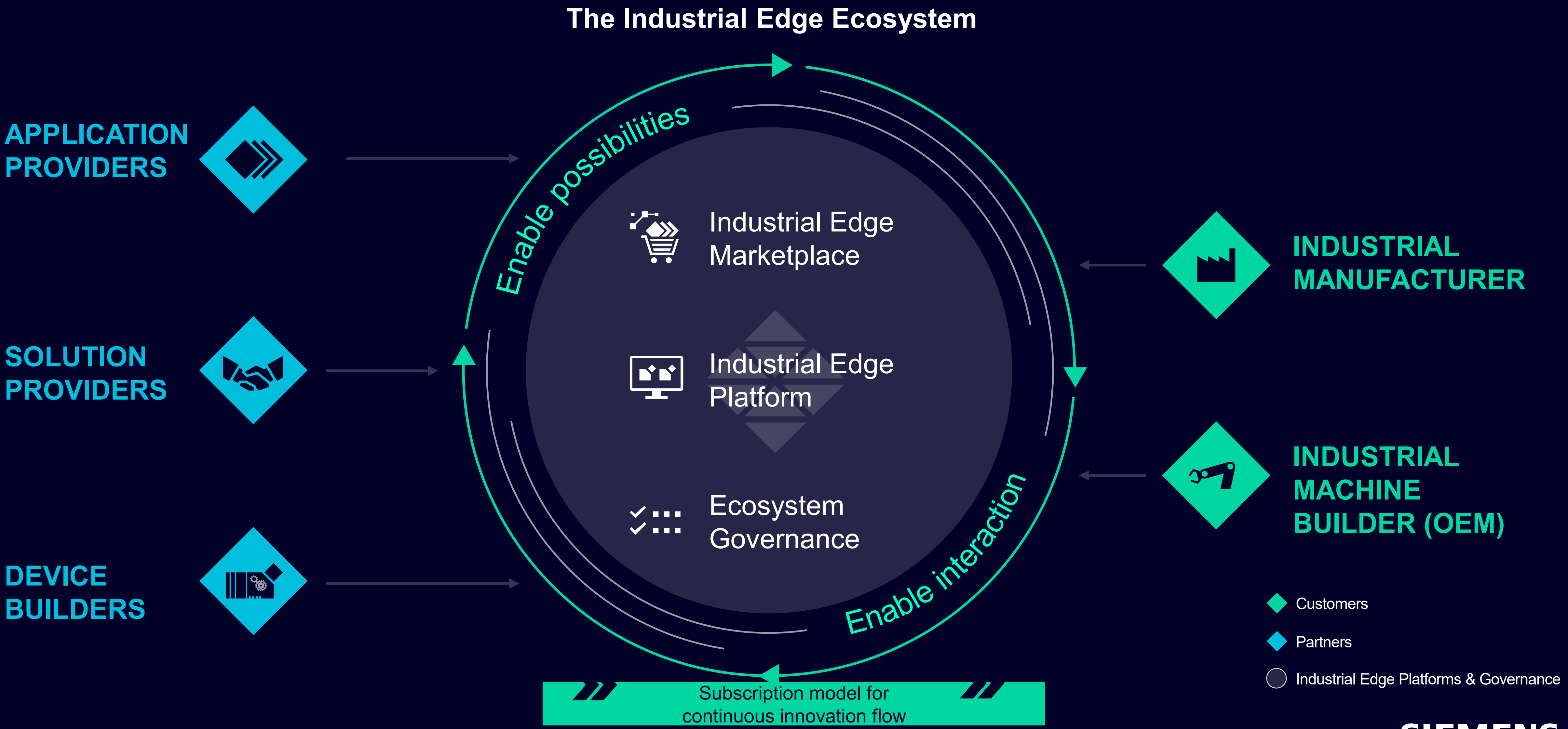
- Implementation Details

Conclusion: Take away & Way Forward

Industrial Edge Ecosystem

Brief Introduction, Challenges, Goals

Big Picture: Industrial Edge Ecosystem



The Industrial Edge Ecosystem

APPLICATION PROVIDERS

SOLUTION PROVIDERS

DEVICE BUILDERS

Enable possibilities

Industrial Edge Marketplace

Industrial Edge Platform

Ecosystem Governance

Enable interaction

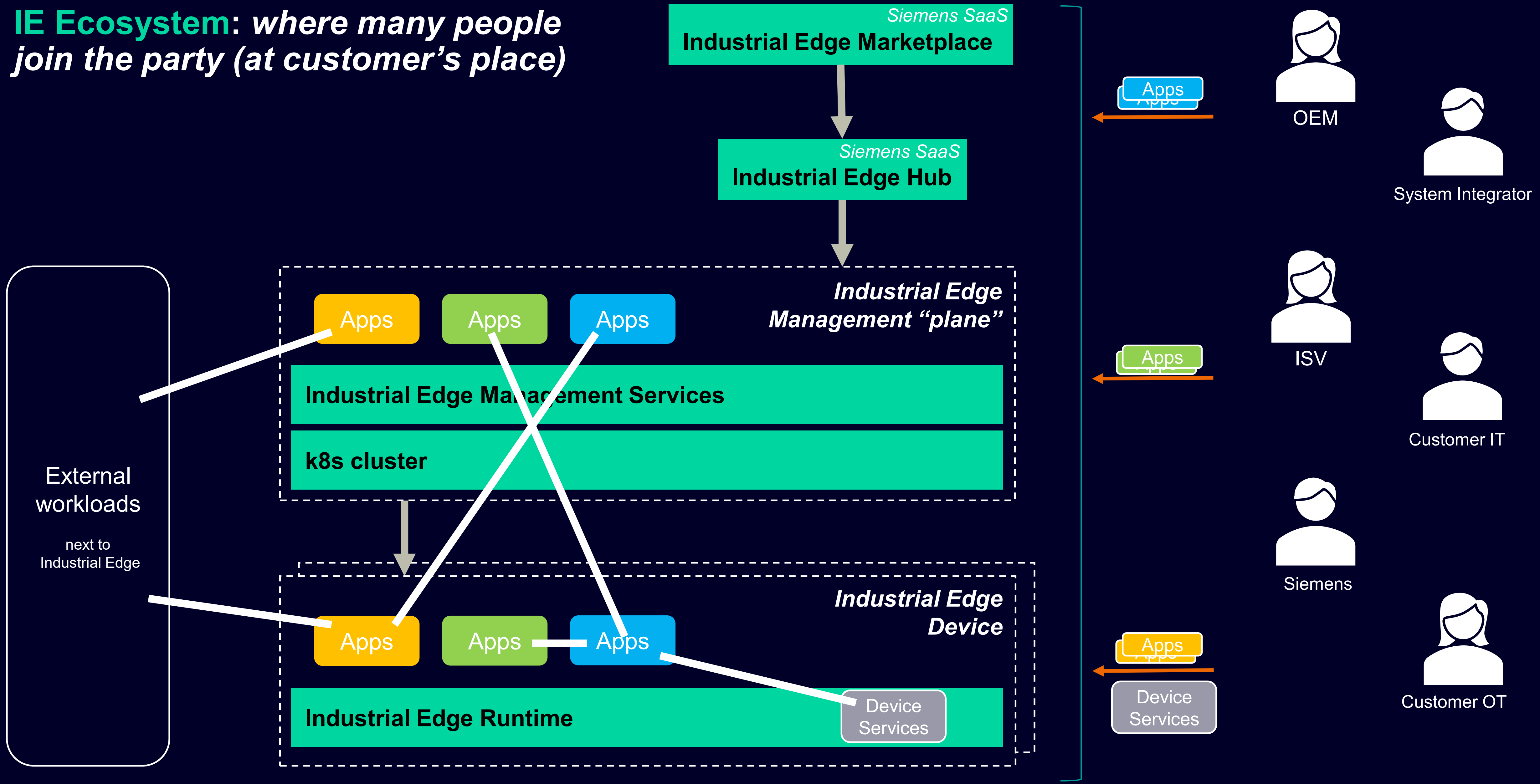
INDUSTRIAL MANUFACTURER

INDUSTRIAL MACHINE BUILDER (OEM)

Subscription model for continuous innovation flow

- ◆ Customers
- ◆ Partners
- ○ Industrial Edge Platforms & Governance

IE Ecosystem: *where many people join the party (at customer's place)*



Challenges & Goals

Zero Trust also a valid concern Industrial Edge scenarios

Securely identify workloads (Edge Applications instances)

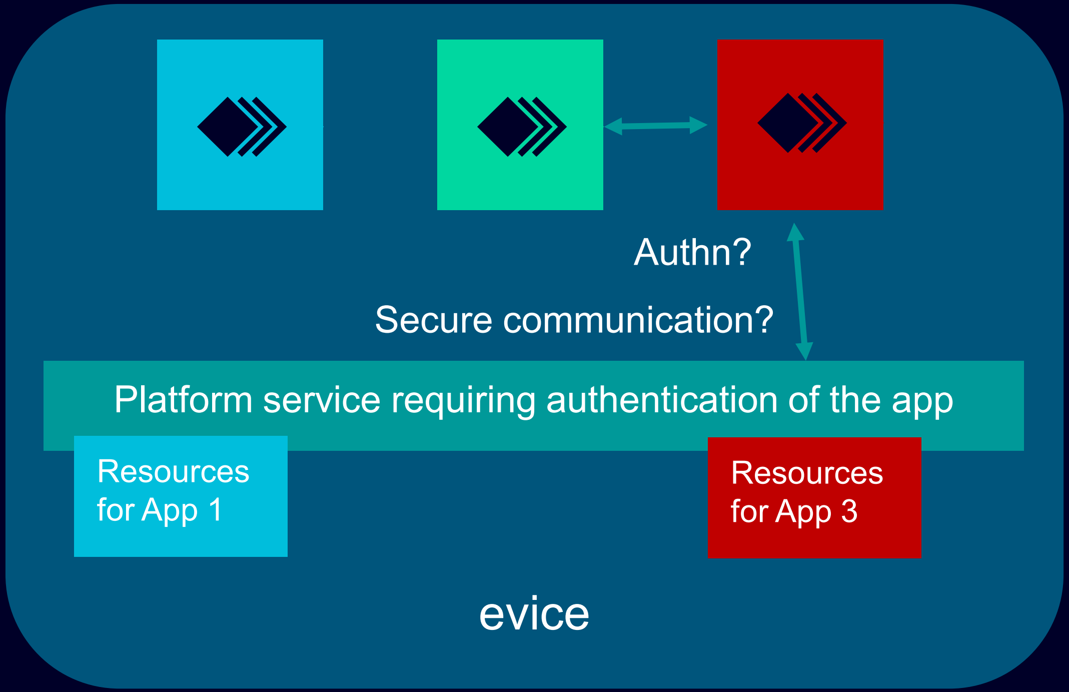
Introduce ideally zero additional effort for Edge Applications

Extensible architecture to enable cross-device/cluster trust

Establish foundation for secure App2App communication

Challenges: Security in App to App Communication

1. Establishing an app Identity on the platform
2. Enabling encrypted and authenticated communication

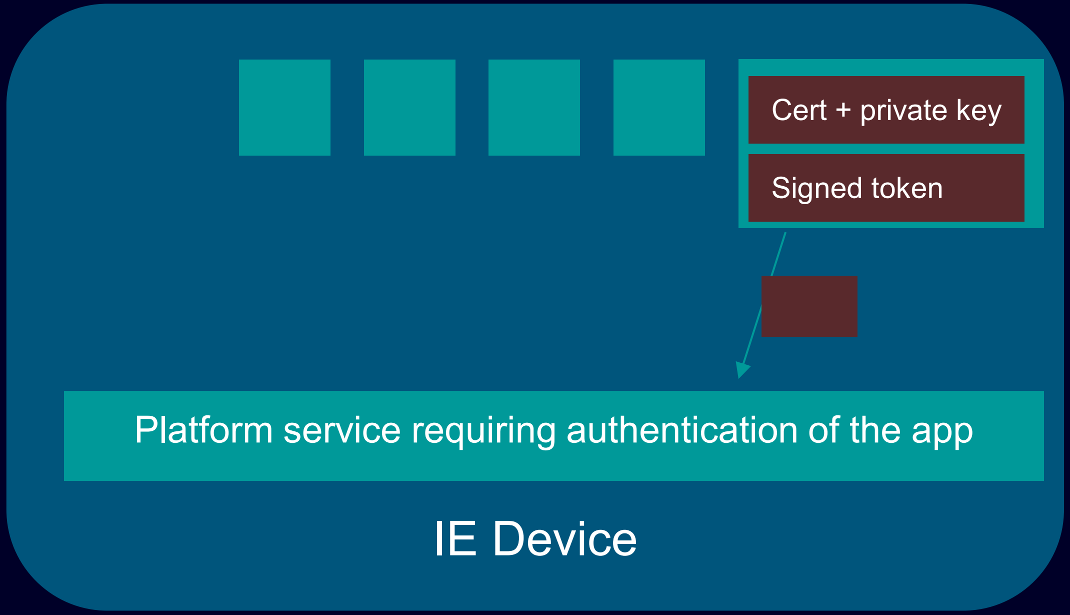


Introduction: spiffe / SPIRE

What is *your Identity?*

Real world

Software defined world



Cert + private key

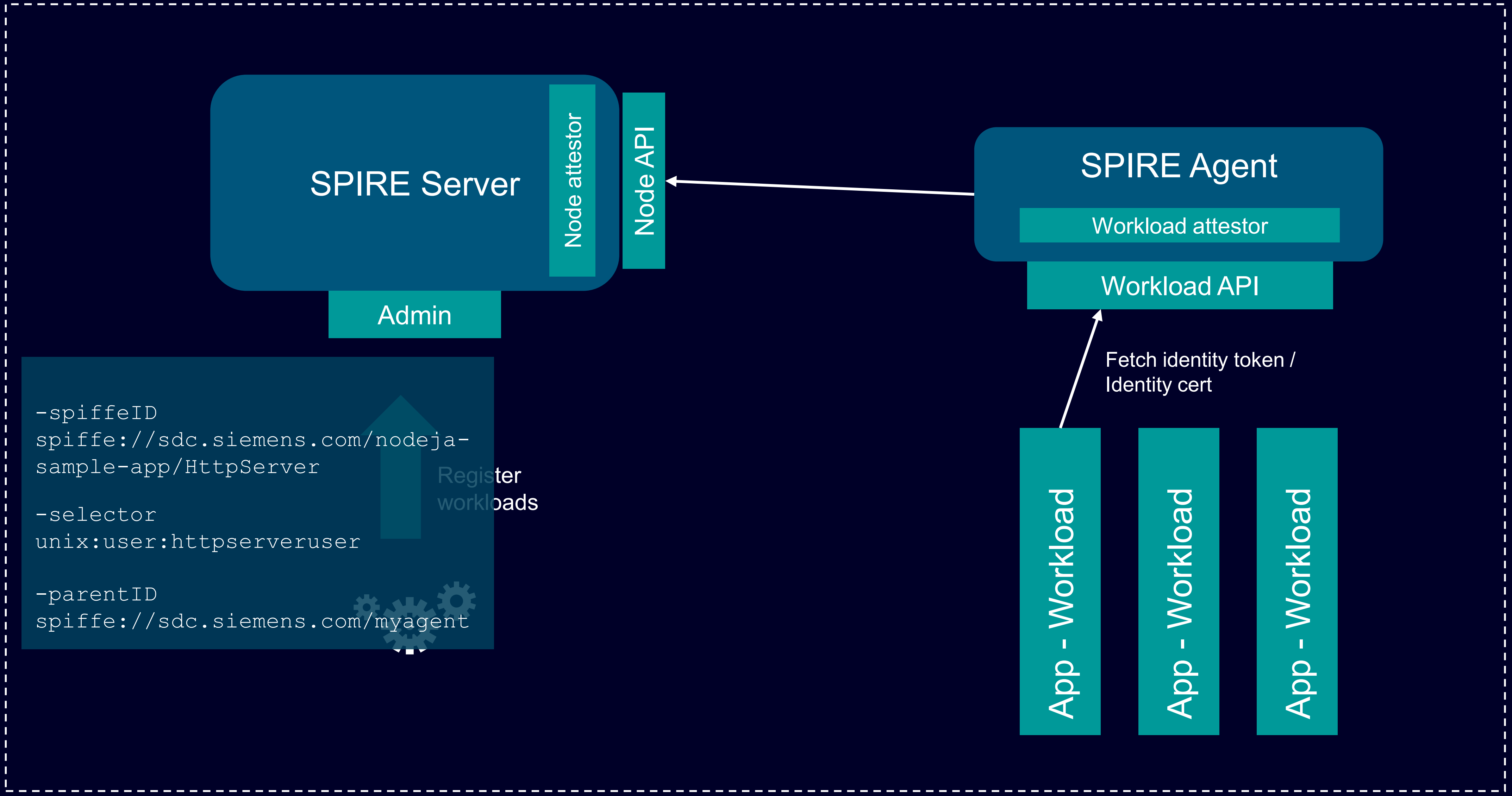
Signed token

Platform service requiring authentication of the app

IE Device

How to get these credentials?

SPIFFE/SPIRE Overview



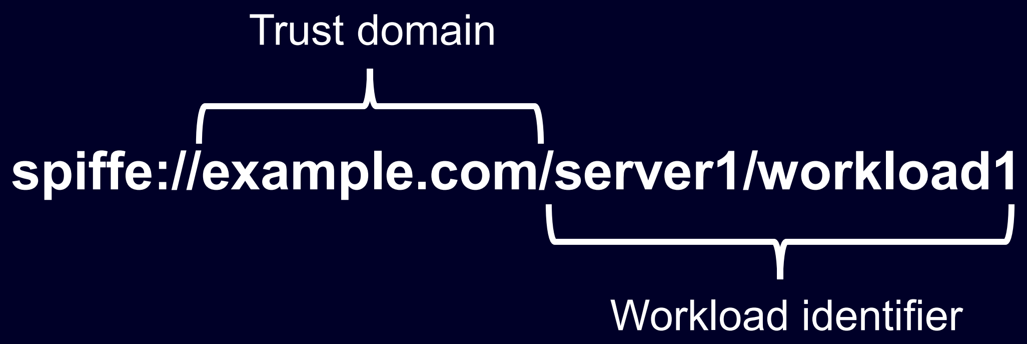
```
-spiffeID
spiffe://sdc.siemens.com/nodeja-
sample-app/HttpServer

-selector
unix:user:httpserveruser

-parentID
spiffe://sdc.siemens.com/myagent
```

SPIFFE SVIDs

eyJhbGciOiJFUzI1NiIsImtpZCI6IjJ1RGtYTXV1eDdaSXJza0RRWEVC
QXVIVWJVMzFmTjhiIiwidHlwIjoiSldUIn0.
eyJhdWQiOlsic3BpZmZlOi8vc2RjLnNpZW1lbnMuY29tL25vZGUtanMtc
2FtcGxlLWFwcC9IdHRwc1NlcnZlciJ
mlhdCI6MTcxNDM3NDkzMCwic3Vi
MuY29tL25vZGUtanMtc2FtcGxlLWF
m8P…Kg

```
{
  "aud": [
    "spiffe://sdc.siemens.com/node-js-sample-
app/HttpsServer"
  ],
  "exp": 1714375230,
  "iat": 1714374930,
  "sub": "spiffe://sdc.siemens.com/node-js-
sample-app/HttpsClient"
}
```

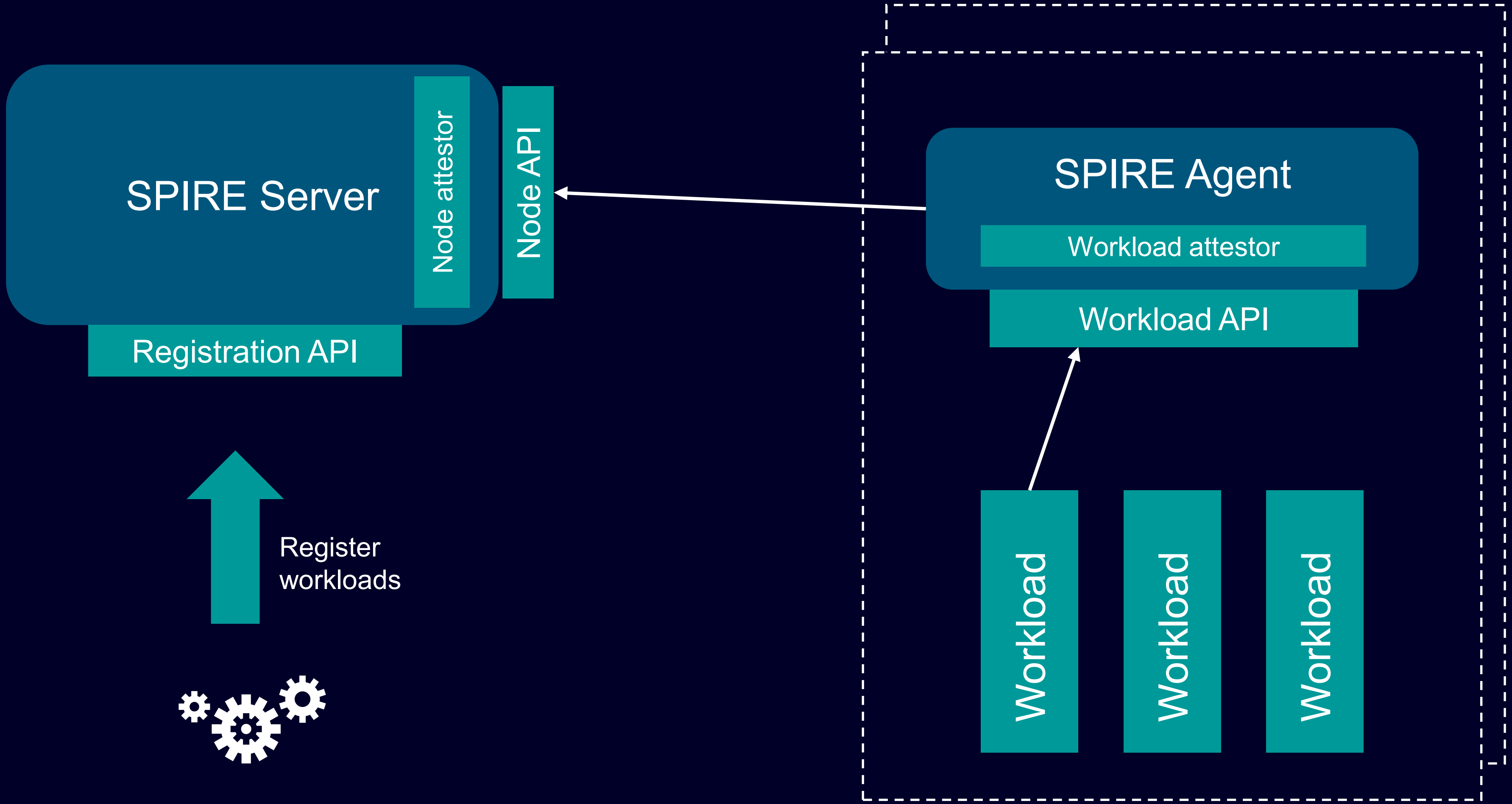
Trust domain



Trust domain

spiffe://example.com/server1/workload1

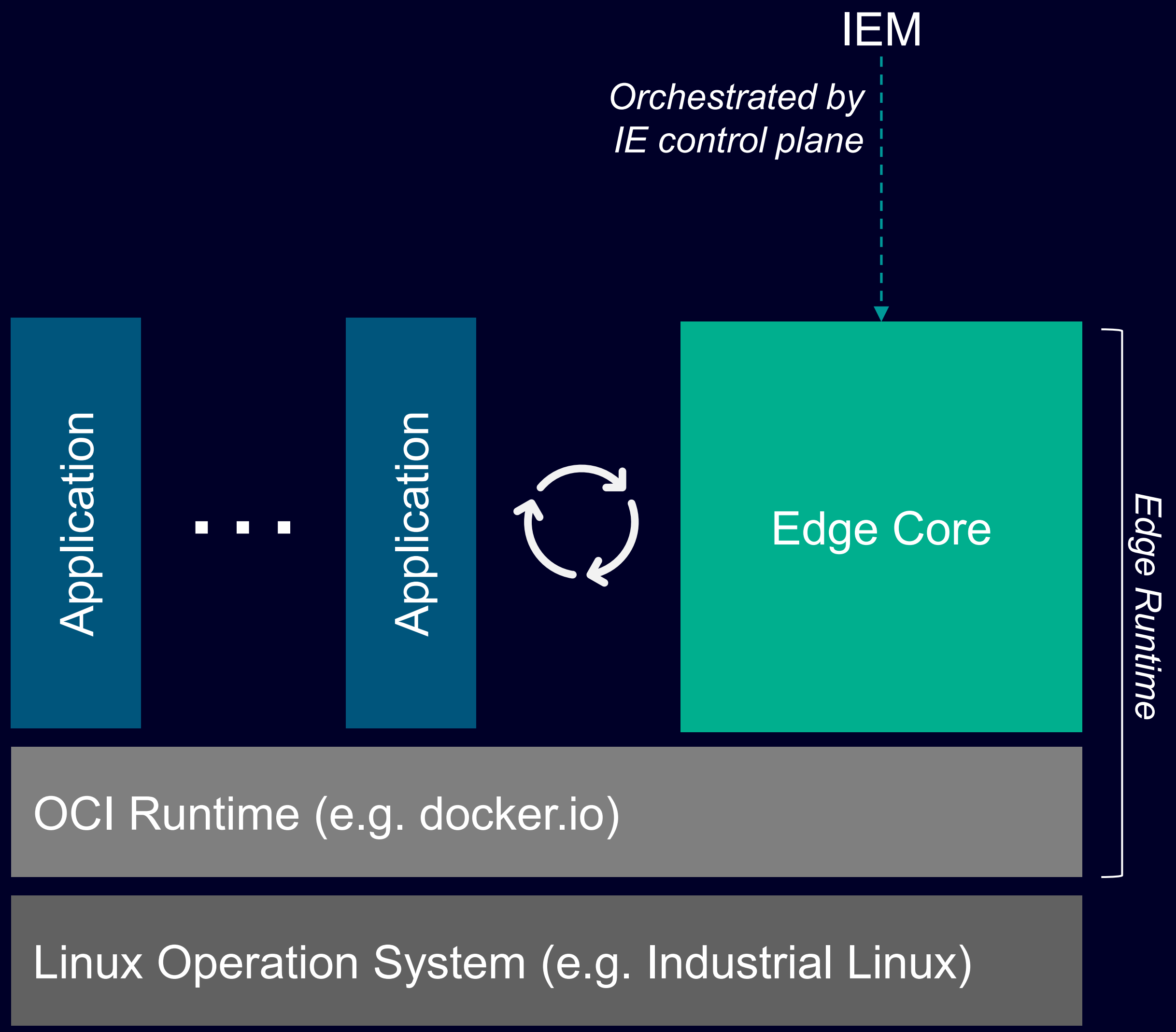
Workload identifier


SPIRE Server
Node attestor
Node API
Registration API
Register workloads
SPIRE Agent
Workload attestor
Workload API
Workload
Workload
Workload

SPIFFE / SPIRE
Adoption within Industrial Edge

Implementation details

Industrial Edge: Application Lifecycle & SPIFFE integration


IEM
Orchestrated by IE control plane
Application
. . .
Application
Edge Core
Edge Runtime
OCI Runtime (e.g. docker.io)
Linux Operation System (e.g. Industrial Linux)

Edge Core

- Overall: manages Lifecycle of Applications (according IEM)
- Utilizes OCI-Runtime APIs to setup, start and stop container instances

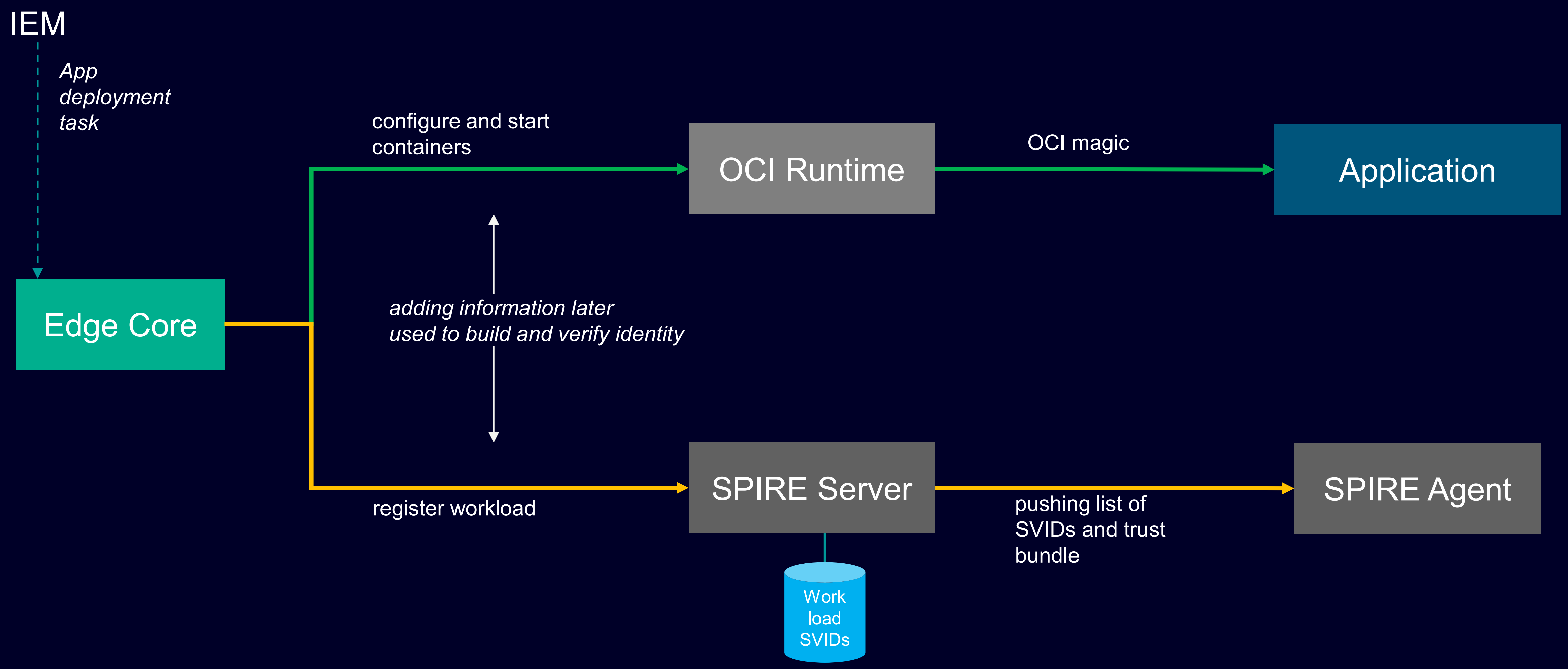
In terms of SPIFFE

- Edge Core → Workload Orchestrator
- Edge Application → Workload
- Edge Core → also a Workload

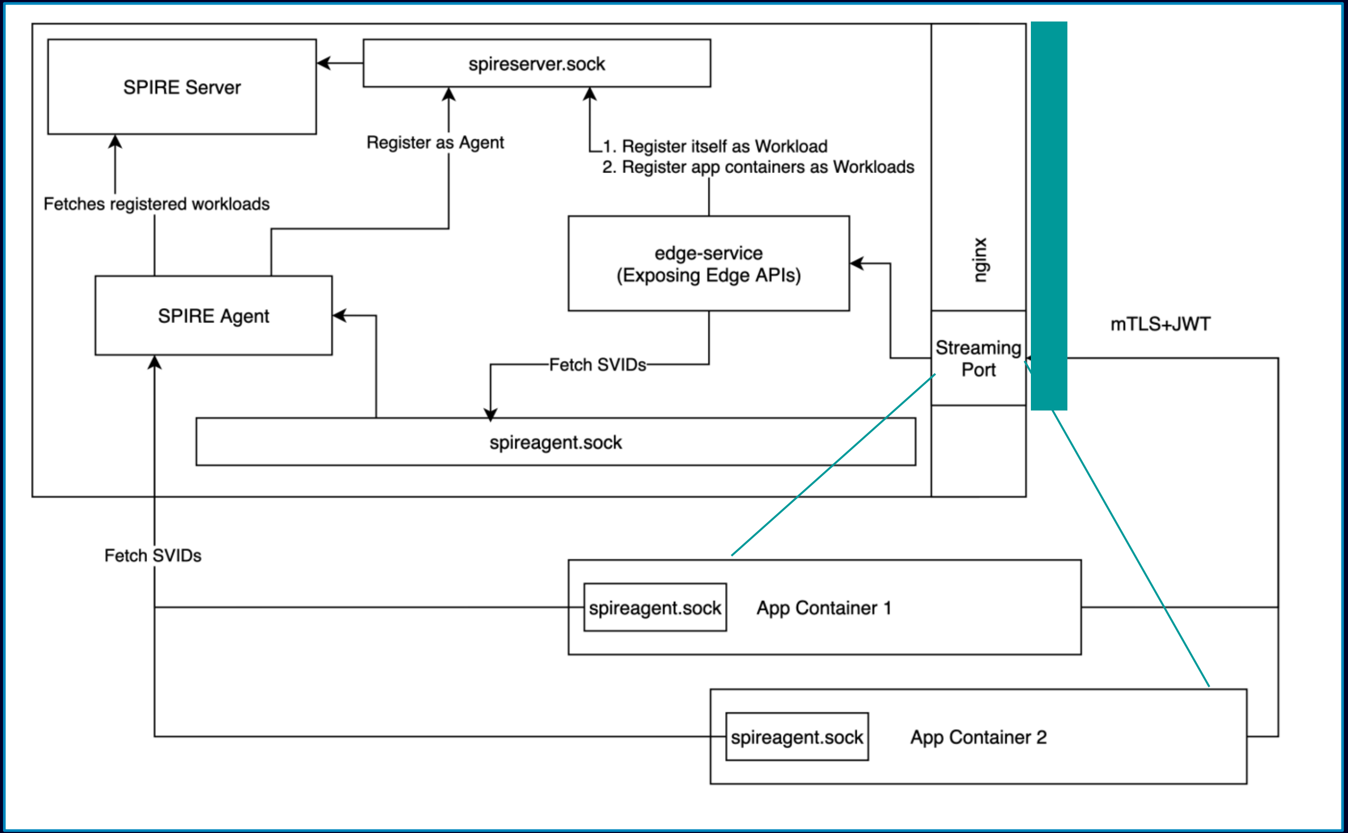
Implementation

- Edge Runtime is extended by a SPIRE-Agent and a SPIRE-Server
- Edge Core* registers Applications using the **Registration API**
- grpc APIs of SPIRE (Agent/Server) exposed via Unix Domain Sockets* and mounted to every container instance

Industrial Edge: Extend Application Lifecycle by SPIFFE “flows”


IEM
App deployment task
Edge Core
configure and start containers
OCI Runtime
OCI magic
Application
adding information later used to build and verify identity
register workload
SPIRE Server
Work load SVIDs
pushing list of SVIDs and trust bundle
SPIRE Agent

SPIFFE/SPIRE integration in Industrial Edge: overall architecture + App PoV



overall architecture

Application POV

- Applications need to handle SPIFFE flows (JWT, X.509 or convenient libraries)
- Edge Platform Services will require SVID
  - edge APIs
  - service registry
  - secure store
- App has to join *proxy-redirect* or host network[1]
- Identity of Agent is injected via EnvVar

➔ public documentation & Example-App will be available soon

[1] not recommended

Conclusion: Take Aways & Way Forward

- Flexibility of SPIRE architecture and available plugins enabled a fast integration into Industrial Edge runtime

- Even if today's implementation within Industrial Edge is not fully leveraging the potential of SPIFFE and SPIRE, we created a future-proof easy to extend security infrastructure

- Potential extensions (not yet decided)
  - extend trust-domain to cross IE-devices by introducing a cluster-wide SPIRE Server and using the UpstreamAuthority ”spire” plugin
  - Adding options to specify custom “Policy” to control ”who can talk to whom”
  - Leverage already established device (birth) certificates (manufacturer certificates) bound to TPM for node attestation
  - Support for mixed infrastructure of Industrial Edge and non-Industrial Edge environments (k8s, <you name it>)

Contacts

Published by Siemens 2024

Sören Stelzer

Principal Engineer for Cloud/Edge Computing & Co-Lead Architect Industrial Edge

soeren.stelzer@siemens.com

Dr. Andreas Reiter

Senior Key Expert – Cybersecurity

andreasreiter@siemens.com