Open APIs for Open Minds

Endpoint-Auth-Service

stefan.wiedemann@fiware.org

FIWARE

Problem description

- NGSI-Brokers support different operations for connecting other systems
  - Notifications - send data via http(or mqtt) so receivers
  - Registrations - request data from other brokers
  - Federation - brokers exchange data

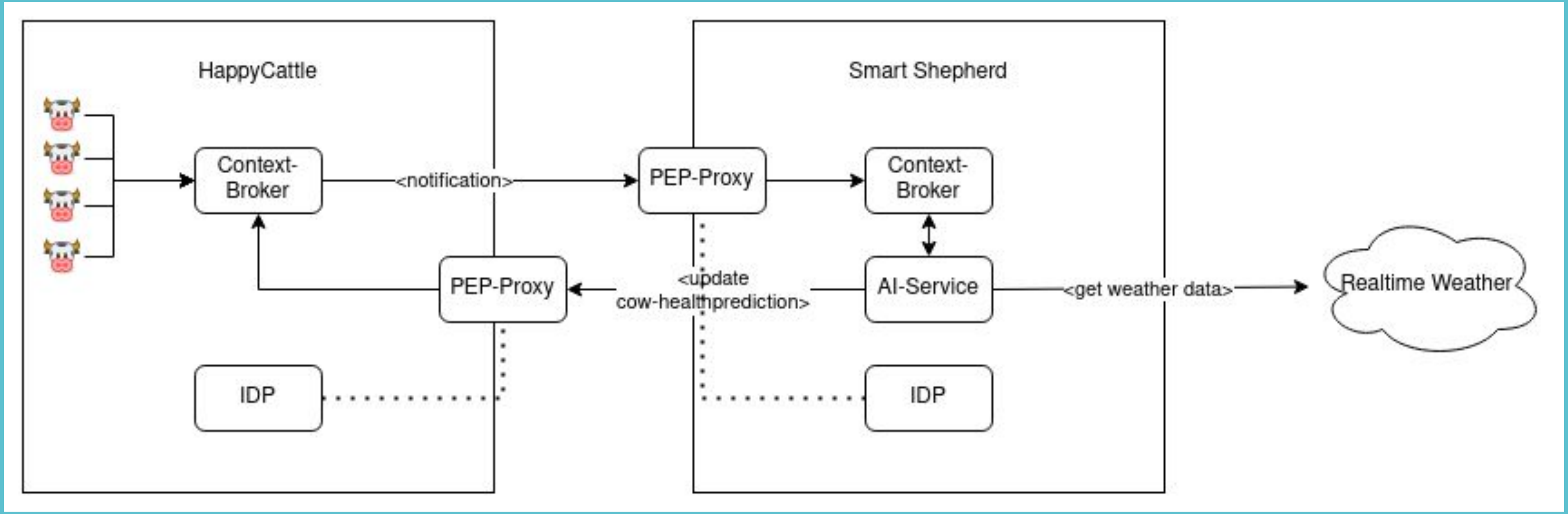
- Connected systems use a security framework for authz/n
  - different methods(oauth2, jwt, cert-auth etc.)
  - different IDPs

- Brokers do not support any auth-method on external calls
  - security is not in the domain of NGSI-LD brokers
  - depending on implementation approach, high effort to implement different methods
  - tight coupling of different concerns(auth & brokering)
  - less extendable

FIWARE

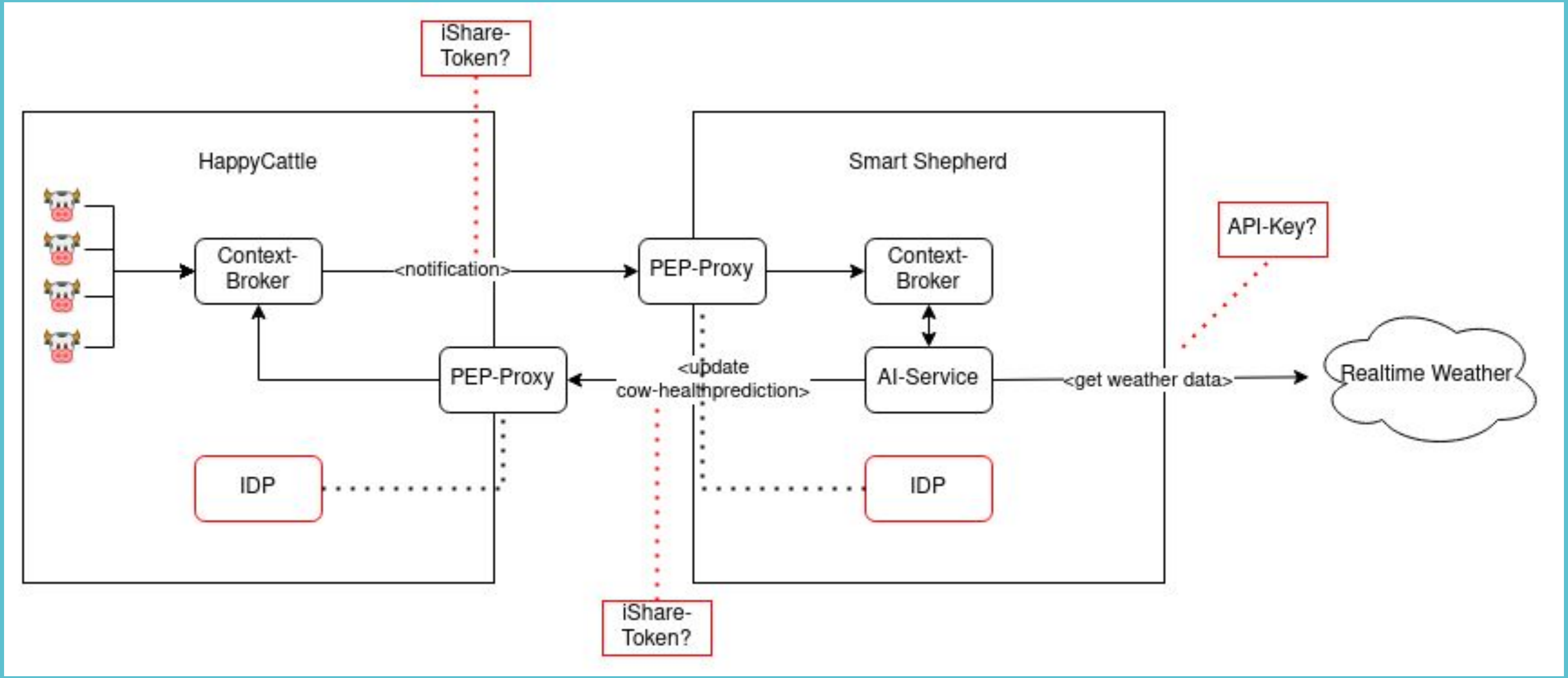
Example: iShare PoC

- HappyCattle sends information about its cattle to SmartShepherd
- SmartShepherd provides AI-Services to predict animal health
- SmartShepherd enriches the data with weather information



Example: iShare PoC

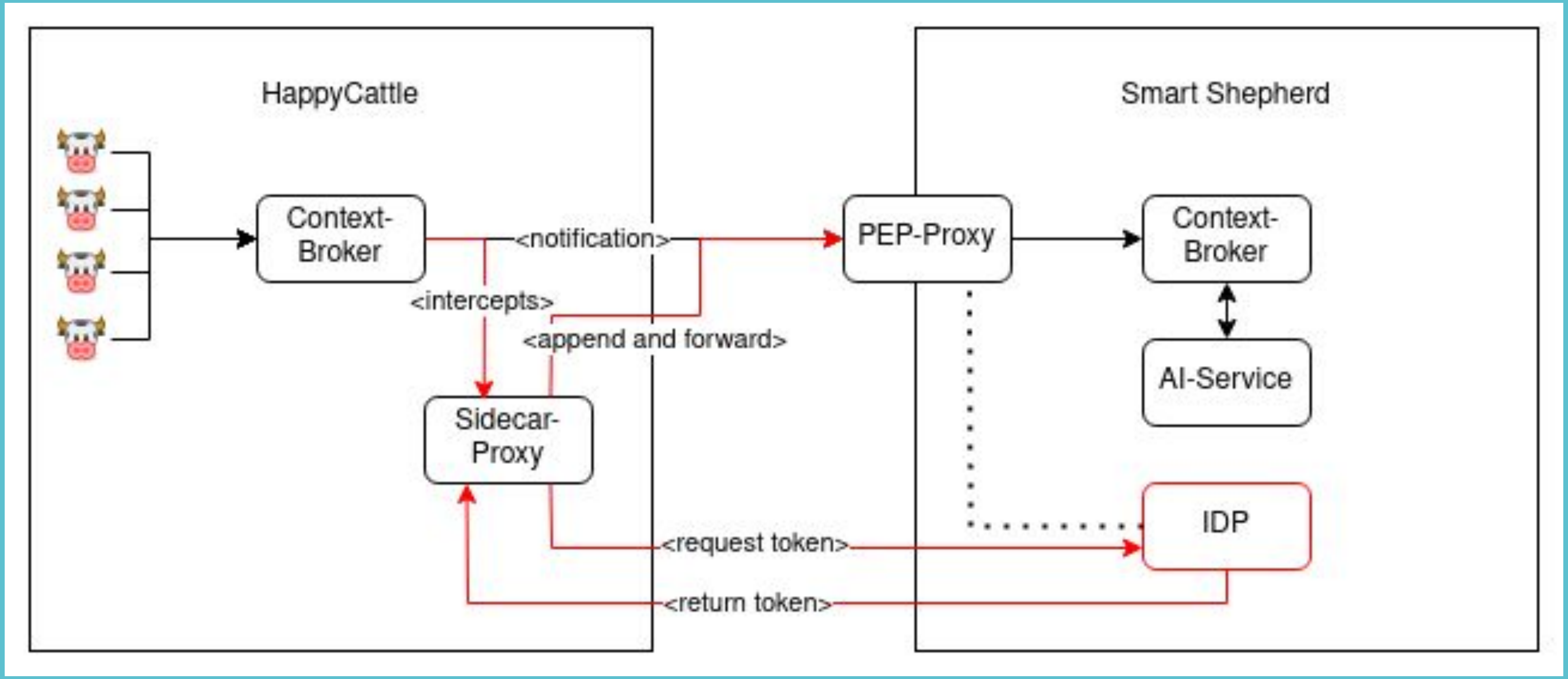
- HappyCattle checks incoming requests (iShare-compliant JWT[1])
- SmartShepherd checks incoming requests (iShare-compliant JWT)
- Realtime Weather checks API-Key



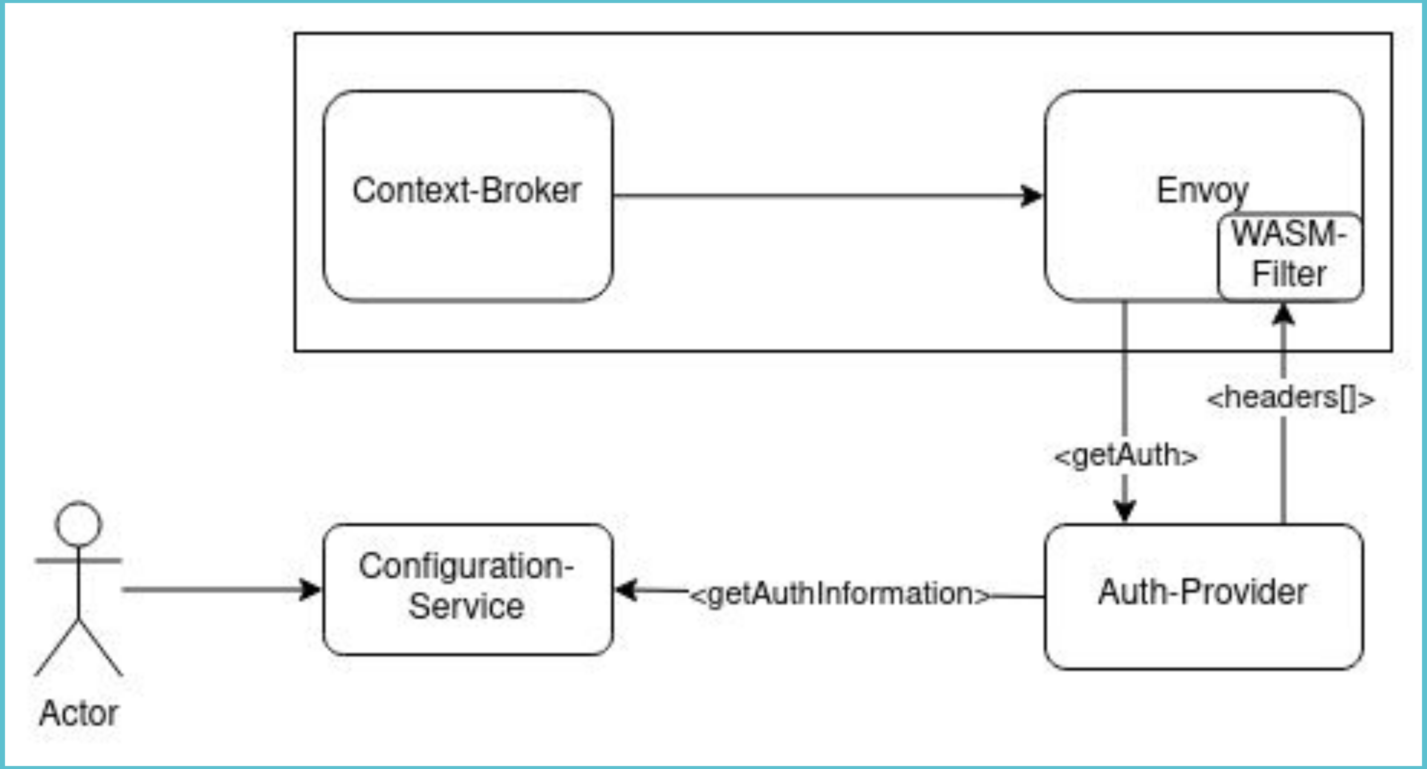
[1]https://dev.ishareworks.org/index.html

Example: iShare PoC

- HappyCattle' Context-Broker sends notification to Smart Shepherd
- Sidecar-Proxy intercepts the request, fulfills authentication-flow and adds the token
- Smart Shepherd PEP-Proxy validates and authorizes the token, forwards request to broker



Architecture Overview



Components - Endpoint-Configuration-Service

- ➢ Github-Repo
- ➢ Configuration-API

- provides configuration capabilities for endpoints(e.g. target-addresses)
  - authType to be applied(currently only iShare implemented)
  - host, port(default: 80) and path(default: /) - all sub-paths will be handled
  - https - should https be used for the request
  - authCredentials - authType specific information about the credentials to be used for the endpoint(f.e. address of IDP, ClientID to be used)
- provides auth-information for the auth-provider to be used
- generates configuration for the envoy-proxy
  - meshExtension: generates ServiceMeshExtension-resources to be consumed by OSSM
  - cluster&listener.yaml: generates the yaml-files to be consumed by envoy

Components - Sidecar-Proxy

➢ Github-Repo

- WASM-Filter for envoy
- can be used as a meshExtension, via aio-container or as a wasm-binary
- handles:
  - decision on which requests to be handled
  - request the configured auth-provider
  - apply returned headers & https
  - cache the headers, depending on the cache-control header from the auth-provider

FIWARE

Components - Auth-Provider

➢ Github-Repo
➢ API

- has to implement a single endpoint
    - takes domain, path and provider-type as input
    - returns a list of header-value pairs to be applied by the proxy
    - provides a cache-control header

- implementation of the concrete auth-mechanism
    - handles the flow to the IDP
    - (potentially) provide an API for managing the auth-information to be used by the provider(e.g. IDs, secrets)

    -> currently only iShare implemented

Components - various supporting components

- init-iptables - sidecar container to apply iptable-manipulation for intercepting the requests
- envoy-configmap-updater - sidecar for config-service to publish cluster&listener.yaml as a configmap to be consumed by envoy
- envoy-resource-updater - sidecar for envoy to read a configmap and copy it to the pod, using a copy-method that triggers the envoy-filewatcher for config-update
- mesh-extension-updater - sidecar for config-service to publish mesh-extension resources via server-side-apply into the mesh
- helm-chart - two modes
    - ossm-integration: functionality will be provided as serviceMesh-extension and injected via mesh-extension-updater
    - sidecar-injection: without OSSM, the proxy can be injected via mutating webhook. A sidecar-injector will be deployed, that watches for annotations on pods to inject the proxy

Github: fiware/endpoint-auth-service

- repo: https://github.com/FIWARE/endpoint-auth-service
- integration-tests: https://github.com/FIWARE/endpoint-auth-service/tree/main/integration-test
- local setup: https://github.com/FIWARE/endpoint-auth-service/blob/main/docker-compose/README.md

Thank you!

http://fiware.org
Follow @FIWARE on Twitter