Security Training

Contrasec

Ilari Mikkonen

Ilari.mikkonen@contrasec.fi

Tarmo Hiltunen

Tarmo.hiltunen@contrasec.fi

Co-financed by the Connecting Europe Facility of the European Union

Table of content

Co-financed by the Connecting Europe Facility of the European Union

Training mandate from GA

“The security toolkit will also include security training for public organisations”

others are free to join too.

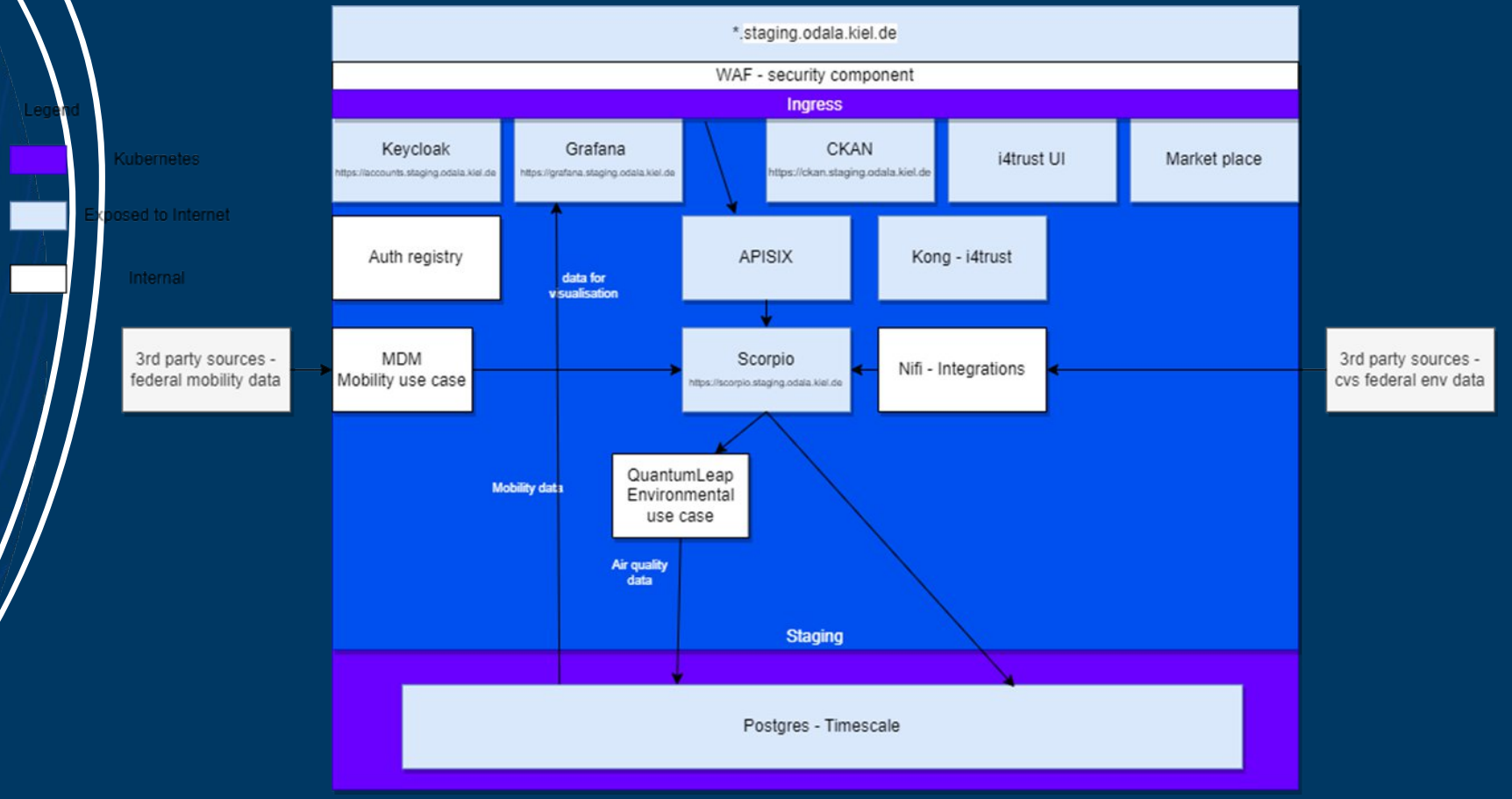
Co-financed by the Connecting Europe Facility of the European Union

Practicalities

1. Do ask questions, but stay in point. If you are not talking, stay muted.
2. You are being recorded. The recording will become public.
3. Session is 2 hours. 50 minutes + 10 minute break is planned + then last 45 minutes + 15 minutes for discussion. We are flexible and may go off script.
4. Follow up sessions can be arranged.

Co-financed by the Connecting Europe Facility of the European Union

Architectural overview



Co-financed by the Connecting Europe Facility of the European Union

Observations and recommendations 1/2

Any open port is a way in. In higher level, ports are then abstracted and used by other components:
- Login screens
- APIS

During the project execution, a security audit was performed. It's a good idea to run sec audits on regular intervals. Good principle is to minimize attack surface by only exposing ports that are absolutely necessary.

Another one is to follow CVE news channel or news letter; in Finland you can subscribe to Traficom's news letter [3]. When a vulnerability is made public or found, analysis of impact and risk needs to be done. After that decision to update components or other action is made.

Document and track. Have a ticketing system where you can track issues and events.

Co-financed by the Connecting Europe Facility of the European Union

Observations and recommendations 2/2

In ODALA we are relying on component providers to update their components so that they are secure. This is often not the case. When selecting the architecture and components for a particular deployment, support for component and evidence of systematic approach to security is one section criteria. In ODALA project this has lead to some decision, like dropping Umbrella from the architecture and providing alternatives via APISIX and Keyrock.

Preventing attacks and making them visible is now possible via the security component
Onion model (or “defence in depth” [1]) is a good approach when designing security; when outer shell is breached, there are still layers in the onion to prevent further advances to the system.

How to operate when you are compromised - it’s important to practice when you have time. Have a plan ready you have practiced, so you don’t have to practice when things have gone wrong.

Be prepared.

Co-financed by the Connecting Europe Facility of the European Union

Holistic security

Security is not a single thing or technical enabler. In practice, holistic security involves integrating multiple security measures and strategies to create a cohesive and robust security framework. This may include a combination of physical security measures such as surveillance systems, access controls, and alarms, as well as digital security measures like firewalls, encryption, and authentication mechanisms. Additionally, it may involve addressing social and psychological factors such as employee awareness, training, and establishing a security-conscious culture within an organization.

By adopting a holistic security approach, organizations and individuals can enhance their resilience against a wide range of threats and vulnerabilities, considering the interconnected nature of modern systems and the diverse ways in which they can be compromised.

Co-financed by the Connecting Europe Facility of the European Union

Holistic security in IT

Holistic security [2] in IT systems is an all-encompassing approach that considers multiple layers of protection, including technology, people, and processes. It integrates technical measures, user awareness, and organizational practices to create a comprehensive defense against cyber threats and vulnerabilities. By taking a holistic approach, organizations can enhance their security posture and better adapt to the evolving IT landscape.

Co-financed by the Connecting Europe Facility of the European Union

Holistic security checklist

1. Secure Kubernetes Configuration:
Implement strong authentication mechanisms for accessing Kubernetes cluster.
Enable RBAC (Role-Based Access Control) to control user permissions.
Regularly update and patch Kubernetes components to address vulnerabilities.
2. API Security:
Implement secure communication channels (HTTPS) for REST API interfaces.
Implement API authentication and authorization mechanisms (e.g., API keys, OAuth).
Apply input validation and sanitize user inputs to prevent common security vulnerabilities such as SQL injection and cross-site scripting (XSS).
3. Network Security:
Employ network segmentation and isolation to restrict unauthorized access.
Configure network policies to control inbound and outbound traffic to and from the cluster.
Implement a firewall or network security groups to filter and monitor network traffic.

Co-financed by the Connecting Europe Facility of the European Union

Holistic security checklist

4. Container Security:
Use trusted container images from reputable sources.
Enable container image vulnerability scanning and regularly update images.
Implement container runtime security measures like resource limits and restrictions.

5. Data Security:
Encrypt sensitive data at rest and in transit using strong encryption algorithms.
Implement access controls and least privilege principles to restrict data access.
Regularly backup data and store backups securely.

6. Monitoring and Logging:
Set up centralized logging to collect and analyze logs from the cluster and API interfaces.
Implement monitoring and alerting systems to detect and respond to security incidents promptly.
Regularly review and analyze logs to identify potential security issues or anomalies.

Co-financed by the Connecting Europe Facility of the European Union

Holistic security checklist

7. Employee Awareness and Training:
Provide security training and awareness programs for employees to promote best security practices.
Emphasize the importance of strong passwords, regular updates, and reporting suspicious activities.
Encourage employees to follow security protocols and report any security incidents.

8. Incident Response and Recovery:

Develop an incident response plan with clear roles, responsibilities, and communication channels.
Conduct regular security drills and testing to validate the effectiveness of the plan.
Define a recovery strategy and backups to restore systems and data in case of an incident.

Co-financed by the Connecting Europe Facility of the European Union

Security Component - why

Deployment is via GitLab CI/CD pipeline - Setup is ready after CI/CD pipeline
Setup can be modified by changing values in CI/CD file

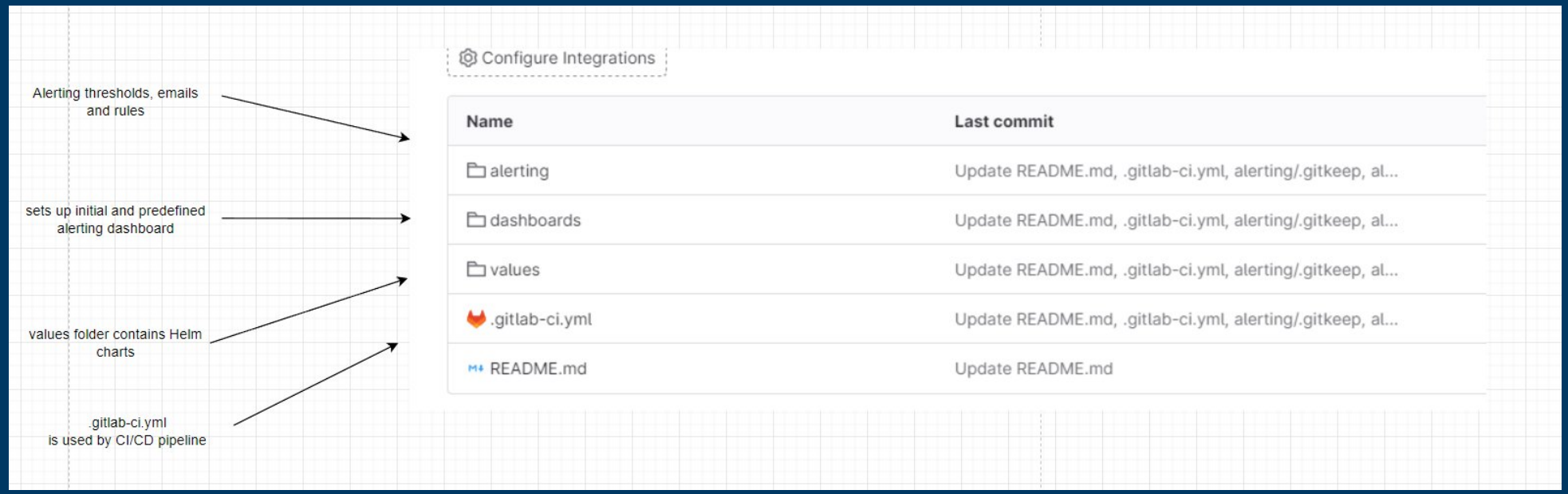
Security Component or WAF (Web Application Firewall) blocks malicious requests by responding HTTP 403 Forbidden message for malicious request - Malicious request will be logged and alert system triggers if alerting thresholds are exceeded

Gathers all security related events from components into one place and therefore minimizes the need of investigating logs from components

Co-financed by the Connecting Europe Facility of the European Union

Security Component - deployment (1/2)

Deployment is via GitLab CI/CD pipeline



Co-financed by the Connecting Europe Facility of the European Union

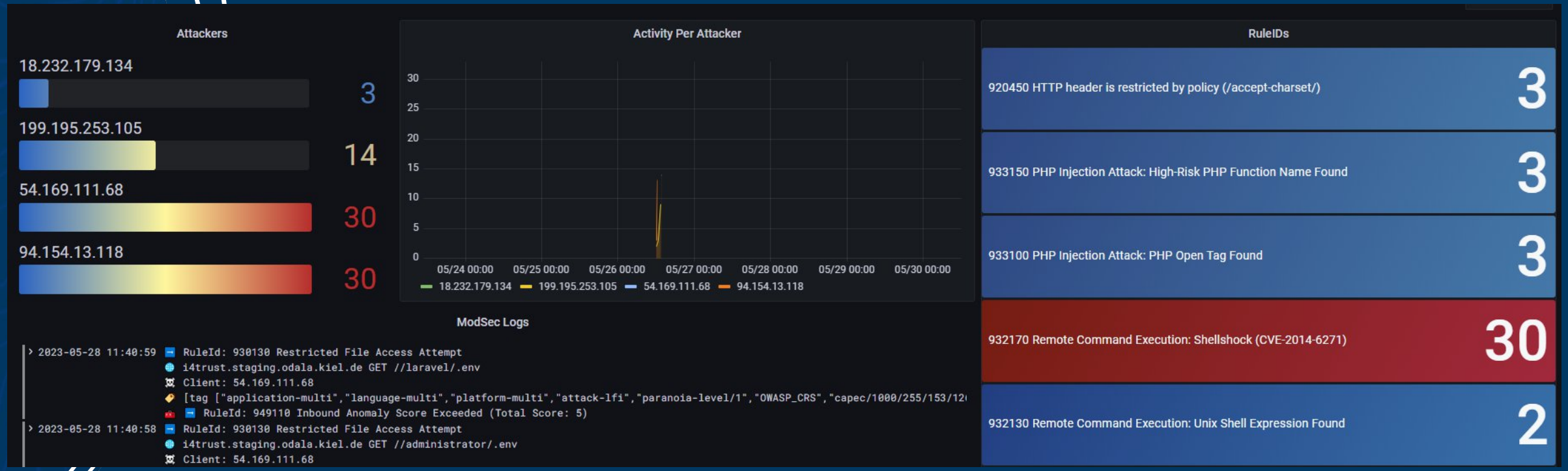
Security Component - deployment (2/2)

Alerting system can be modified from .gitlab-ci.yml file
Threshold variable defines the amount of attacks to trigger alerting
Timerange variable tells alerting system to gather attacks from defined timerange
EmailAddresses variable is used to define the email addresses where alerts will be sent

```yaml
variables:
  KUBE: "$KUBECONFIG"
  # SMTP variables
  EMAILSECURITYCOMPONENTHOSTPORT: $EMAIL_SECURITY_COMPONENT_HOST_PORT
  EMAILSECURITYCOMPONENT: $EMAIL_SECURITY_COMPONENT
  EMAILSECURITYCOMPONENTUSER: $EMAIL_SECURITY_COMPONENT_USER
  EMAILSECURITYCOMPONENTPASSWORD: $EMAIL_SECURITY_COMPONENT_PASSWORD
  # Alert variables
  THRESHOLD: 10 # Number of attacks per IP addresses to alert
  TIMERANGE: 30 # Minutes | Defines the time range from now and defined timerange
  EMAILADDRESSES: "tarmo.hiltunen@contrasec.fi;" #$EMAILSECURITYCOMPONENT # Send alerts with email.
  # Other variables
  GRAFANA_ADMIN_PASSWORD: $SECURITY_COMPONENT_GRAFANA_PASSWORD
```

Co-financed by the Connecting Europe Facility of the European Union

Security Component - Dashboard

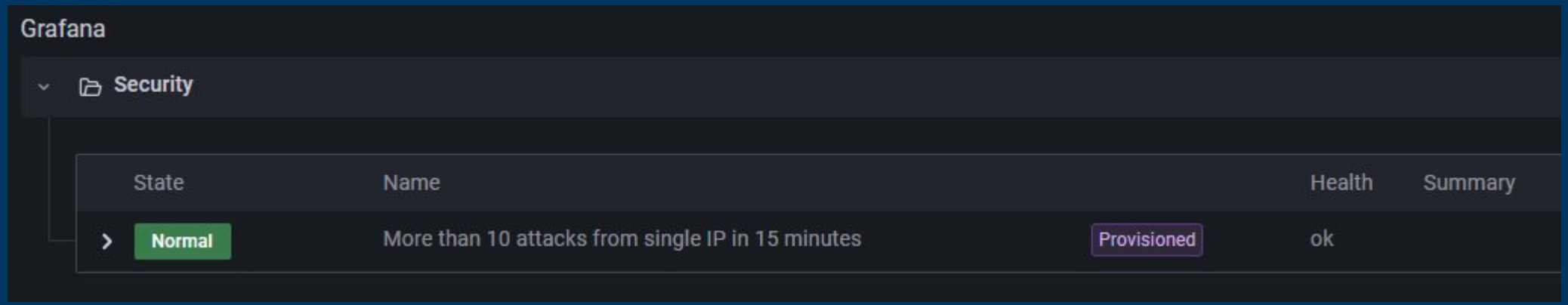


Co-financed by the Connecting Europe Facility of the European Union

Security Component – alerting

One alert rule is pre-defined for deployment
Alert will be sent if there are more than X attacks from single IP address in defined timerange

Let’s demo



Co-financed by the Connecting Europe Facility of the European Union

APISIX - deployment and integration

Deployment is via GitLab CI/CD pipeline.
Follows similar pattern like other components, Helm charts and substituting values to template files.
For example, APISIX has an admin API. Using that API we setup predefined routes and auth plugin (scorpio-client).

```
"client_id": "scorpio-client",
"discovery": "http://keycloak/realms/apisix-realm/.well-known/openid-configuration",
"grant_type": "urn:ietf:params:oauth:grant-type:uma-ticket",
"http_method_as_scope": false,
"keepalive": true,
"keepalive_pool": 5,
"keepalive_timeout": 60000,
"lazy_load_paths": false,
"permissions": [
  "res:scorpio#scopes:read-data"
]
```

```
# APISIX
## Wait for apisix
- until nc -z $APISIX_HOST $APISIX_PORT 2>&1; do sleep 10; done
## Configure Apisix with Scorpio route
- curl $APISIX_HOST:$APISIX_PORT/apisix/admin/routes/1 -H "X-API-KEY:$APISIX_API_KEY" -X PUT -T ./apisix/routes/scorp
## Configure Apisix with crowdflowobserved route
- curl $APISIX_HOST:$APISIX_PORT/apisix/admin/routes/1001 -H "X-API-KEY:$APISIX_API_KEY" -X PUT -T ./apisix/routes/cr
## Configure Apisix with noiselevelobserved route
- curl $APISIX_HOST:$APISIX_PORT/apisix/admin/routes/1002 -H "X-API-KEY:$APISIX_API_KEY" -X PUT -T ./apisix/routes/no
## Configure Apisix with trafficflowobserved route
- curl $APISIX_HOST:$APISIX_PORT/apisix/admin/routes/1003 -H "X-API-KEY:$APISIX_API_KEY" -X PUT -T ./apisix/routes/tr
## Configure Apisix with weatherobserved route
- curl $APISIX_HOST:$APISIX_PORT/apisix/admin/routes/1004 -H "X-API-KEY:$APISIX_API_KEY" -X PUT -T ./apisix/routes/we

- echo variables $APISIX_HOST $SCORPIO_INTERNAL
```

Co-financed by the Connecting Europe Facility of the European Union

Keycloak- deployment and integration

Deployment is via GitLab CI/CD pipeline.
Follows similar pattern like other components, Helm charts and substituting values to template files.
We manually configured a client in Keycloack, and then imported that using Keycloak's import mechanism.
It's recommended to follow this pattern, or then make regular backups of the database (postgres).
User creation and management can be done via CI/CD pipe, but this is quite cumbersome and recommend manual approach, if there are not too many user requests.

Co-financed by the Connecting Europe Facility of the European Union

APISIX - configuration model

Crash course to APISIX concepts [4]:

Route: A route is a routing path to upstream targets. In Apache APISIX, routes are responsible for matching client's requests based on defined rules, loading and executing the corresponding plugins, as well as forwarding requests to the specified upstream services. In APISIX, a simple route can be set up with a path-matching URI and a corresponding upstream address.

Upstream: An upstream is a set of target nodes with the same work. It defines a virtual host abstraction that performs load balancing on a given set of service nodes according to the configured rules.

If any changes need to be done to APISIX configuration, we recommend doing it via CI/CD pipe. This way the changes are under version control.

Co-financed by the Connecting Europe Facility of the European Union

Keycloak - configuration model

Basic stuff like demo users, realms etc. can be done via CD/CD pipe. Application in Keyclock is quite complex and requires a lot of details to be handled.

We decided to manually configure a client and then import that using Keycloak's import function.

We recommend making changes by hand or via CI/CD.

We recommend doing user management manually.

Co-financed by the Connecting Europe Facility of the European Union

APISIX & Keycloak - operations

APISIX should not require many operations during life cycle. But if it’s not fire and forget, then we recommend to do the operations via CI/CD pipe.

Let’s look at the CI/CD pipe.

Keycloak may be more dynamic, especially when user management is concerned.

Let’s add a user.

Let’s fetch a token and get some data from Scorpio.

Co-financed by the Connecting Europe Facility of the European Union

Secure data access via marketplace principles

Marketplace is bundled with i4trust plugin - provides i4trust related functions into marketplace

Plugin utilizes third-party components from iSHARE, Authorization Registry and Satellite. Authorization Registry holds organization defined access policies and Satellite holds information about participants / organizations

Plugin allows sellers to monetize NGSI-LD services which customers (organizations) can buy from marketplace

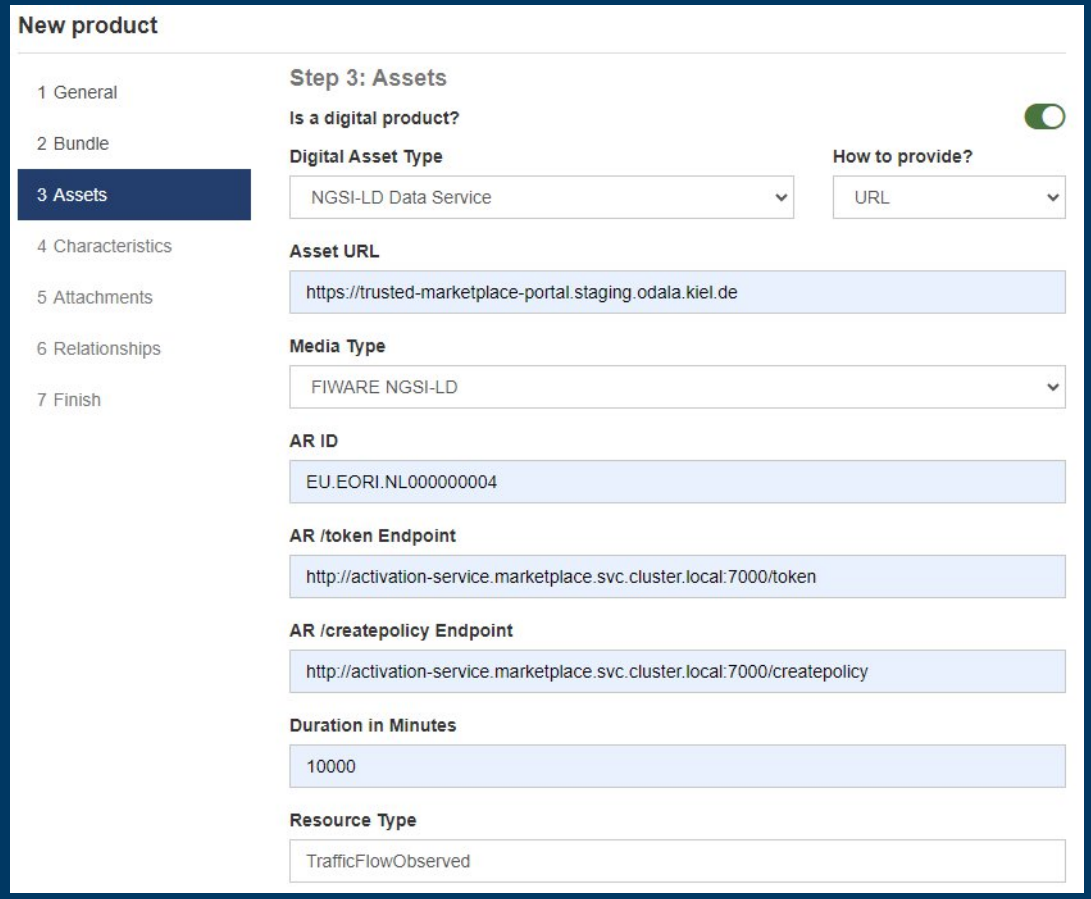
Co-financed by the Connecting Europe Facility of the European Union

Marketplace as a seller

Seller creates a product that is combined with i4trust plugin

Seller defines an access policy and AR (Authorization Registry) configuration for product

Access policy for customer is created when product is sold



ODALA

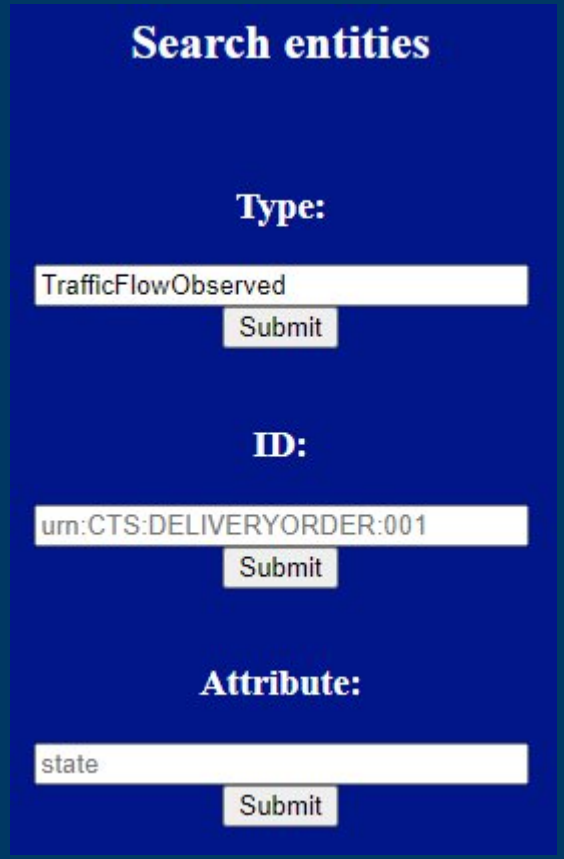
Co-financed by the Connecting Europe Facility of the European Union

Marketplace as a customer

Customer buys the NGSI-LD service from seller

After acquiring the NGSI-LD service, customer can access the data from portal

DEMO Seller and Customer roles



ODALA

Co-financed by the Connecting Europe Facility of the European Union

Umbrella and Keyrock

“API Umbrella” [5], or “Umbrella” is an NREL project. It has then been forked few times, most notable by APInf [6] and FIWARE [7]. In the discussions with Umbrella community, it was discovered that no one is willing to commercially support Umbrella. This was the catalyst to replacement work.

In any case, Umbrella and Keyrock combination is working in Kiel. Umbrella was the proxy being used in i4Trust architecture in the beginning, but now it has been superseded by Kong.

Keyrock is still the default IDM in i4Trust.

Let’s look at the documentation.

Let’s get some data from Scorpio.

Co-financed by the Connecting Europe Facility of the European Union

Documentation

Documentation can be found from there:

https://gitlab.publiccode.solutions/odala-public

ODALA

Co-financed by the Connecting Europe Facility of the European Union

Links and references

[1] https://en.wikipedia.org/wiki/Defense_in_depth_(computing)
[2] https://www.techtarget.com/whatis/definition/holistic-security?utm_campaign=20170320_Word%2520of%2520the%2520Day%3A%2520holistic%2520security&utm_medium=email&utm_source=NLN&track=NL-1823&ad=913416&src=913416&asrc=EM_NLN_74409241
[3] https://www.kyberturvallisuuskeskus.fi/en/ncsc-news/subscribe-our-newsletters
[4] https://apisix.apache.org/docs/apisix/getting-started/configure-routes/
[5] https://github.com/NREL/api-umbrella
[6] https://github.com/apinf/apinf-umbrella
[7] https://github.com/FIWARE/api-umbrella

Co-financed by the Connecting Europe Facility of the European Union

Contact

Ilari.mikkonen@contrasec.fi
contrasec.fi
+358 44 712 6866

Co-financed by the Connecting Europe Facility of the European Union