Introduction

Contrasec is responsible in the ODALA project to implement the security layer; it has been done using Umbrella & Keyrock. In addition it was during project duration decided that Contrasec will implement the Market place as well.

Umbrella is an open source component, reverse proxy and API management. It has been maintained by the community in recent years, but now the support has been fading.

Please note that unexpected leaves on Contrasec side caused the analysis to be rather shallow on technical details.

There are several aspects to this:

1) API management and security layer. At the moment we are with Umbrella and Keyrock. This needs replacing and is simplest of these tasks.
2) Market place. I4Trust framework has support for Market place.
3) Federation and dataspaces.

Options

1) Study replacement components for Umbrella. Short list at the moment is: Apache APISIX and Kong. Reimplement the security layer with selected component. Impact: Estimate is that this will take 20 PD to implement. We can still reach the project goals, but with lesser room for surprises.

2) Do nothing. Impact: City of Kiel cannot go to production after the project ending using the current components, unless someone picks up support for Umbrella and this is not in sight.

Questions and answers

Can Kong support market place via i4Trust – yes, according to FIWARE.

How mature Kong – i4Trust integration is – not sure, but not production ready at this time and not battle tested. This will hopefully change during the i4Trust project in next 12 months.

Why not use one API management only? We have concerns regarding Kong's business model and openess. It may close / change. Apache APISIX is more open, it's in Apache foundation.

Recommendation

Option 1. Implement API management with Apache APISIX and i4Trust with Kong. Contrasec team is most confident with Apache APISIX and can get help from the ecosystem. Kong effort can be hopefully supported via FIWARE and i4Trust. Details:

- First phase redo security layer parts (replace Umbrella) with Apache APISIX and do it as infrastructure as code from the beginning. This allows us to move to a state where Kiel is in secure footing the fastest.
- Do market place with Kong if there is support for that so that we can meet project goals. If not, revert to Umbrella + market place.
- For dataspaces (and federation) use Kong as there should be complete reference implementation available from i4Trust.
- See in the end if there is budget left and port the i4Trust plugins for Apache APISIX. If there is budget left in the project, study how to Reimplement market place with selected component.

Running both Kong and Apache APISIX will allow us to gather experience and ultimately select the best option.

Analysis on the alternatives

Umbrella is not even on the list; No support renders it obsolete.

Both Apache APISIX and Kong are modern alternatives, which boast cloud native capabilities. Both can do infrastructure as code and have NGINX at heart. Both have plugin architecture.

Reason why we picked Apache APISIX as alternative to Kong is that Kuopio's Smart City platform provider selected Apache APISIX as opposed to Kong and we have some experience with it. There were other reasons, but not being able to run everything on-prem (think secure usage and edge applications here) made it an impossible (cost wise) choice. Kiel region has also been working with Apache APISIX. If you want to run on prem, you need Kong Konnect licensing.

Apache APISIX

Pros

- Commercial support available
- New kid on the block
- Contrasec is more familiar with this.
- Used in Kiel region
- Reported performance is higher than Kong
- Kuopio's decision to favour Apache APISIX

Cons

- New kid on the block

Kong

Pros

- Commercial support available
- Widely used
- I4trust support

Cons

- Control plane on-prem only in Enterprise version (Kong Konnect). This is costly.
- From engineering team: Kong was difficult to set up.

Remaining budget – Contrasec

By end of August, Contrasec 65 PD (25 PD are ear marked for Market place) remaining on the Activity 2 – security layer. This means effectively we have 40 PD left. This will be consumed during the project execution, given that we still have not finalised the Data spaces and federation work. Effectively we have no budget left for hardening and productization.

7,5 PD on Activity 3 – Front runner city implementation. This is not sufficient for any development.

Impact on budget

Item	Cost in PDs (one PD is 8 hours)	Notes
Kong study	2	spent

API SIX study	1	Spent
Implement selected API management solution with Apache APISIX	25	Rough estimate
Fix and maintain Umbrella	N/A	Not feasible, we do not have the necessary skills nor the business justification
Port exiting i4Trust plugins to Apache APISIX	24	Both have similar technologies, should be pretty straight forward.
Market place with BAE and Kong	25	Already factored in, no impact.

Keyrock

I4Trust uses Keyrock. There is some work done on Keycloak side by i4Trust, but it is not complete. As Keyrock implements standard openid connect, we should be good. We should be able to migrate to Keycloak, when so decided.

Dataspaces and Federation

Technical feasibility work has been completed. This work is now on paused, as Tarmo is on a leave until end of October (estimate).

Impact for the follower cities

None of the follower cities in the ODLA project are greenfield cities. This has become clean in the discussions with the follower cities conducted by Contrasec. The meeting minutes are available in cloud. Given this, this decision has no direct impact on the follower city implementation.

It may be that the follower cities will later adopt the security layer, or parts of, developed in ODALA project.

Sources and material

https://medium.com/@ApacheApache APISIX/why-do-you-need-apache-Apache APISIX-when-you-have-nginx-and-kong-4aa3403a053b

https://Apache APISIX.apache.org/

https://api7.ai/blog/api-gateway-apache-Apache APISIX-and-kong-selection-comparison/

https://github.com/FIWARE/tutorials.Securing-Access-OpenID-Connect

https://github.com/FIWARE/catalogue#context-dataapi-management-publication-and-monetization

https://github.com/i4Trust/tutorials/tree/main/PacketDelivery-ReferenceExample/Data-Service-Provider