Moving beyond HTTP

Surveying the state of L7 protocols in the Cloud Native ecosystem

Presented by Jonathan Beri

About me

Product guy

Ex-Google/Nest, startups

Founder @ IoT startup

@beriberikix

Typical IoT platform

Device Messaging

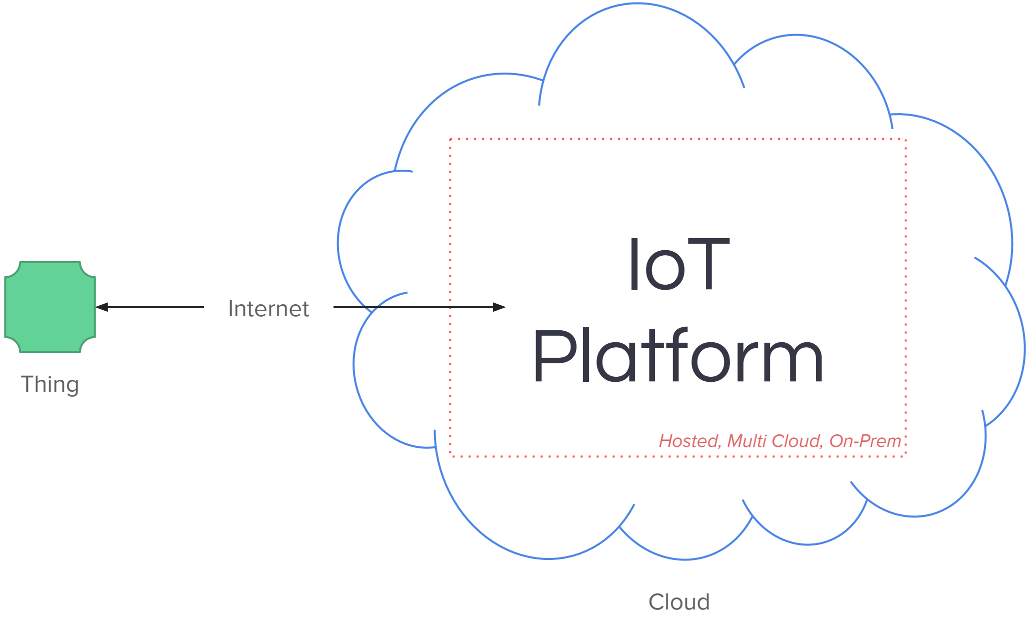
Security

Software Updates

Fleet Management

Compute & Storage

Carrier integrations



IoT protocols

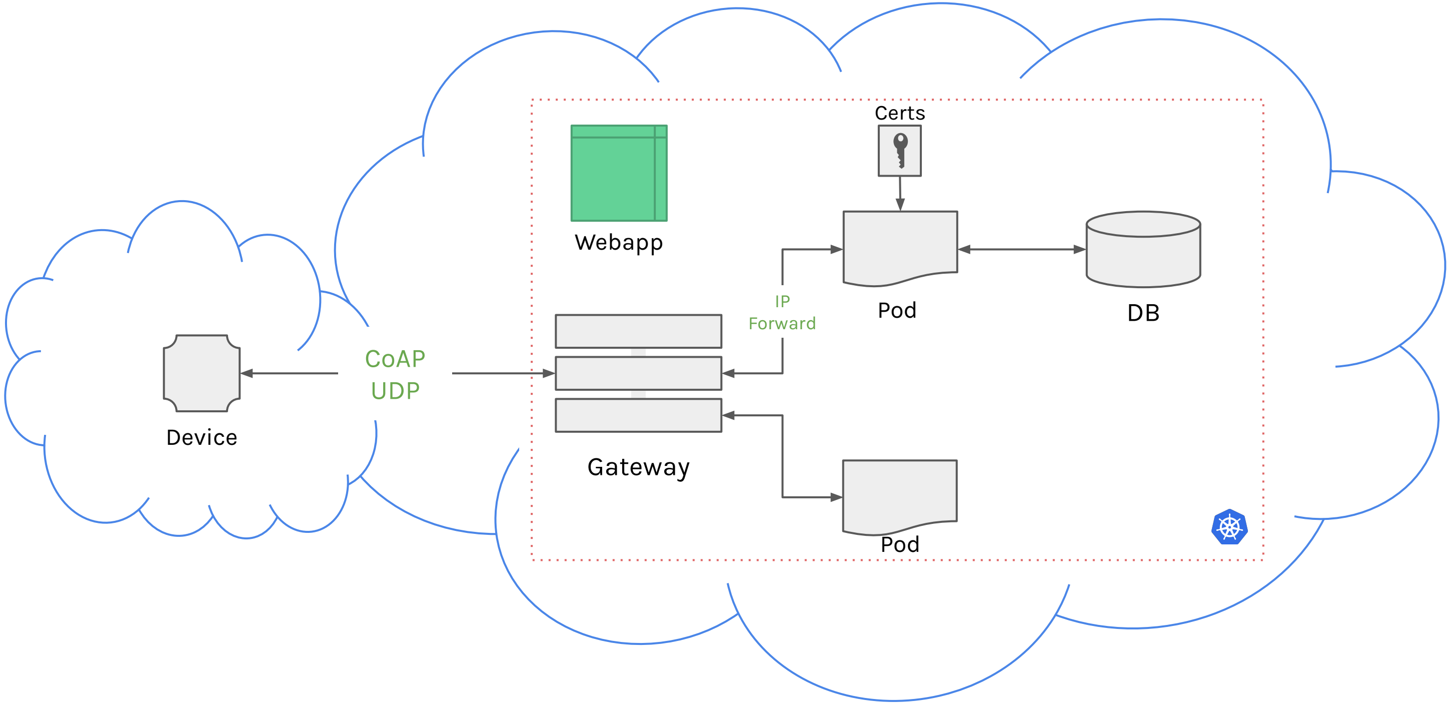
CoAP

LightweightM2M
specified at oma

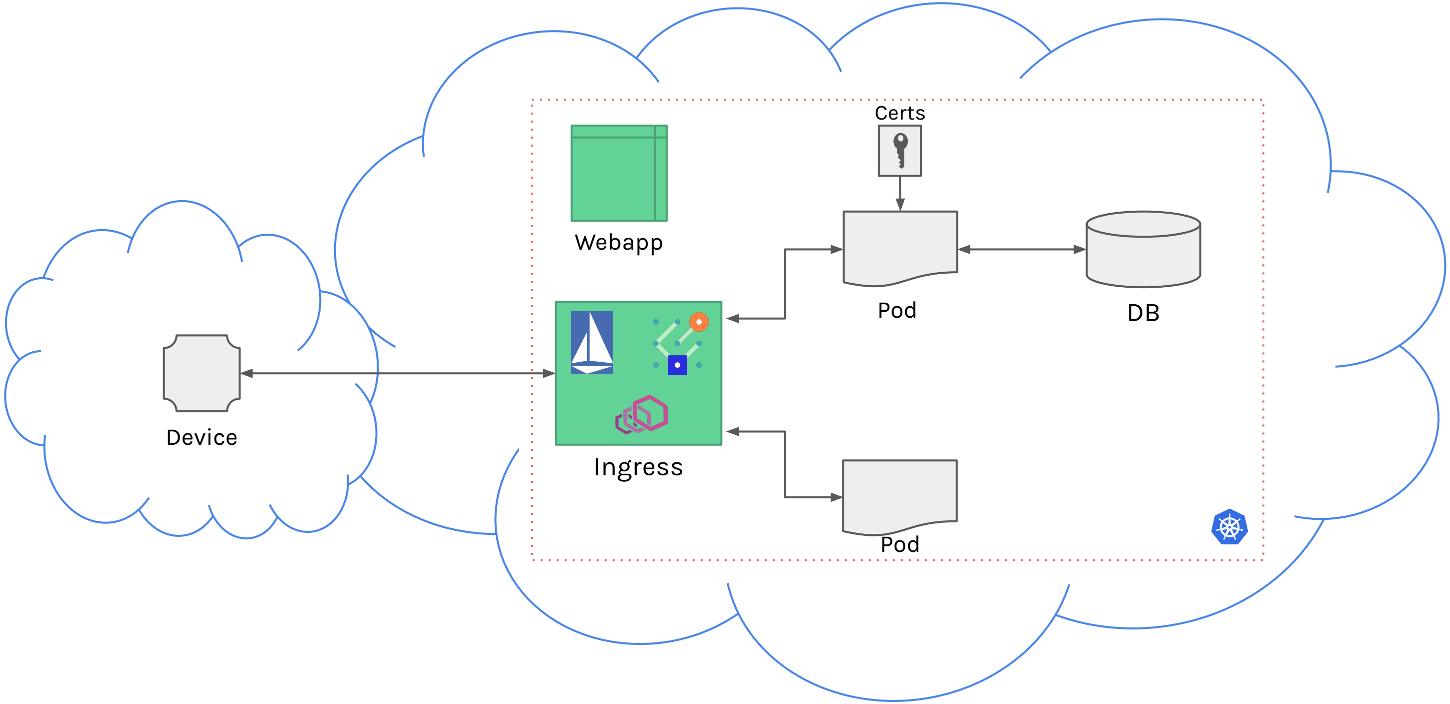
CAN Bus

ASHRAE
BACnet®

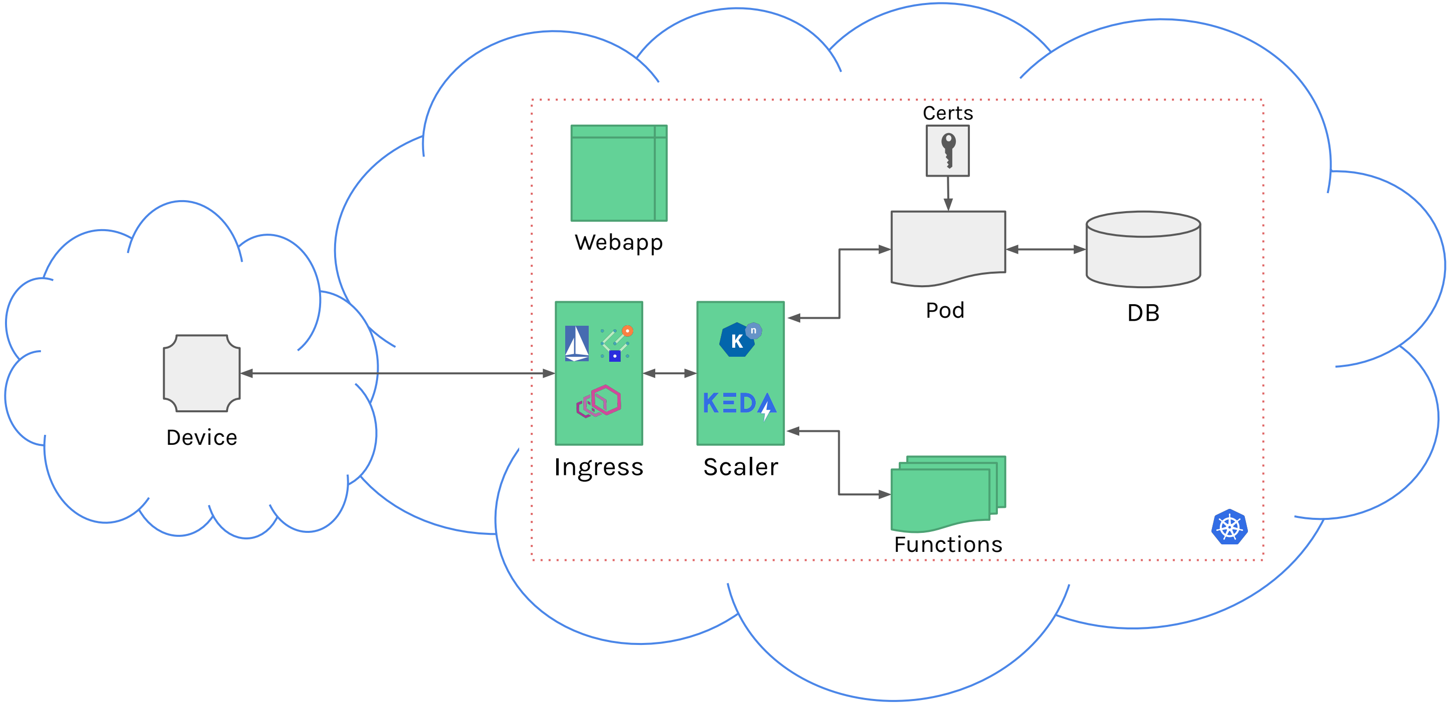
Initial Architecture



Cloud Native

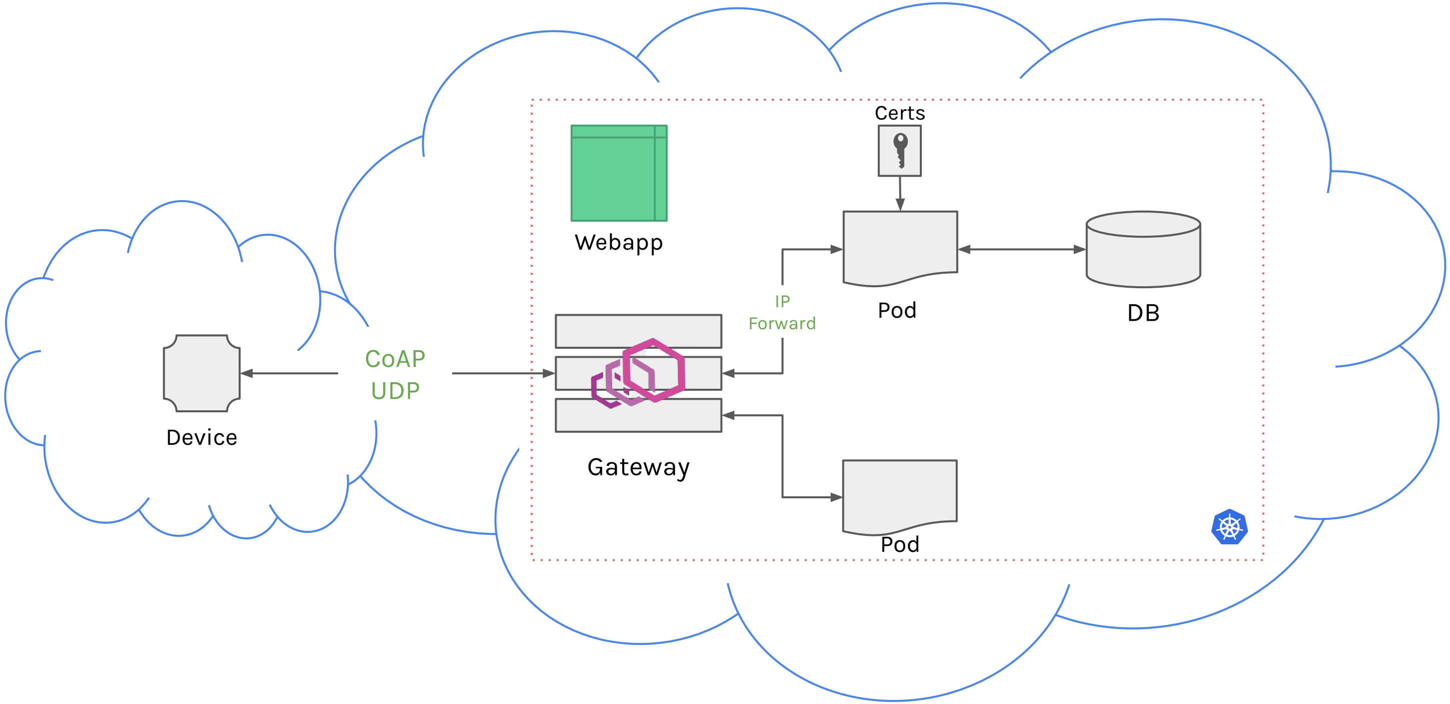


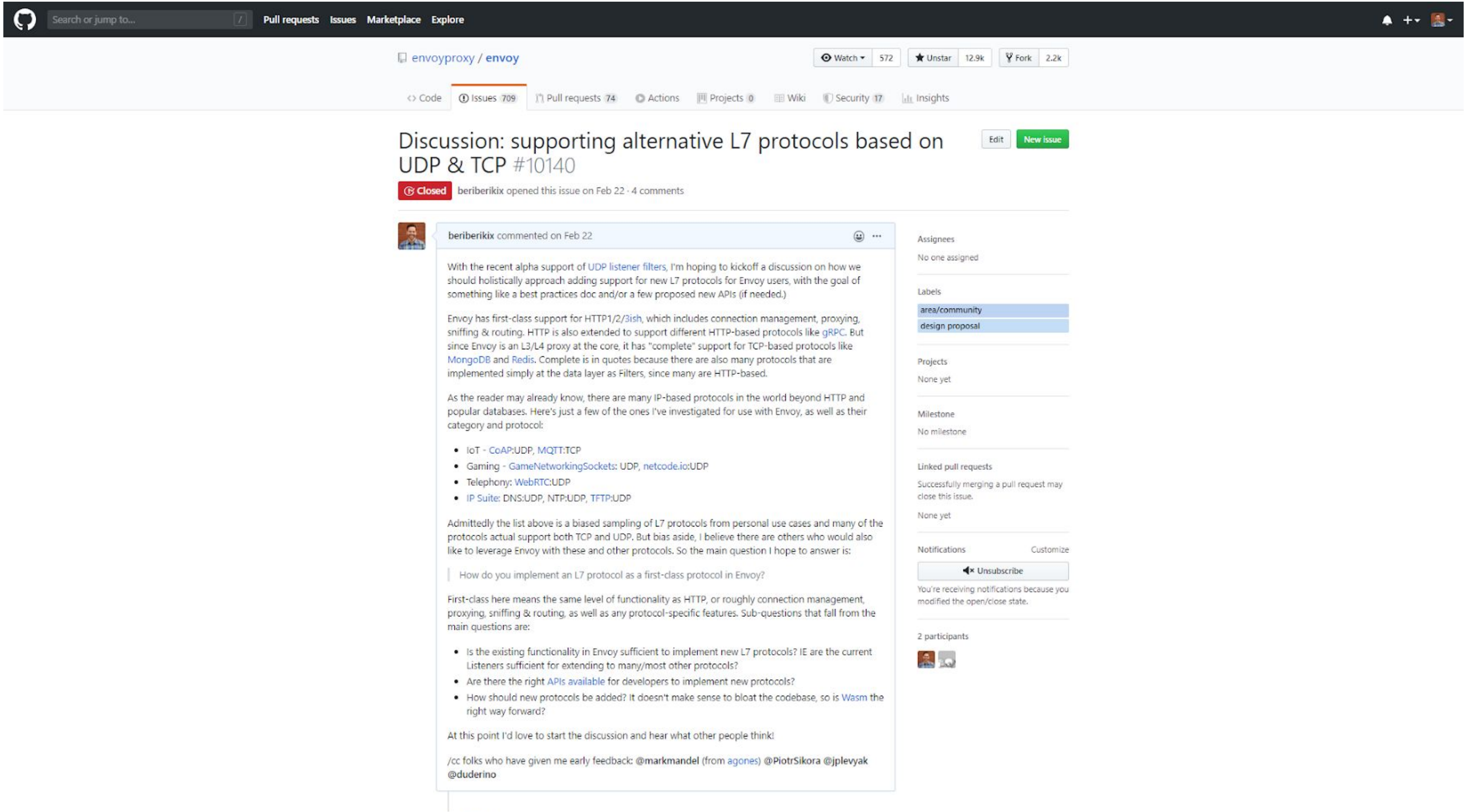
Serverless



Device
Webapp
Certs
Pod
DB
Ingress
Scaler
KEDA
K
n
Functions

First step: gateway + Envoy





Discussion: supporting alternative L7 protocols based on UDP & TCP #10140

Closed beriberikix opened this issue on Feb 22 · 4 comments

beriberikix commented on Feb 22

With the recent alpha support of UDP listener filters, I'm hoping to kickoff a discussion on how we should holistically approach adding support for new L7 protocols for Envoy users, with the goal of something like a best practices doc and/or a few proposed new APIs (if needed.)

Envoy has first-class support for HTTP1/2/3ish, which includes connection management, proxying, sniffing & routing. HTTP is also extended to support different HTTP-based protocols like gRPC. But since Envoy is an L3/L4 proxy at the core, it has "complete" support for TCP-based protocols like MongoDB and Redis. Complete is in quotes because there are also many protocols that are implemented simply at the data layer as Filters, since many are HTTP-based.

As the reader may already know, there are many IP-based protocols in the world beyond HTTP and popular databases. Here's just a few of the ones I've investigated for use with Envoy, as well as their category and protocol:

- IoT - CoAP:UDP, MQTT:TCP
- Gaming - GameNetworkingSockets: UDP, netcode.io:UDP
- Telephony: WebRTC:UDP
- IP Suite: DNS:UDP, NTP:UDP, TFTP:UDP

Admittedly the list above is a biased sampling of L7 protocols from personal use cases and many of the protocols actual support both TCP and UDP. But bias aside, I believe there are others who would also like to leverage Envoy with these and other protocols. So the main question I hope to answer is:

> How do you implement an L7 protocol as a first-class protocol in Envoy?

First-class here means the same level of functionality as HTTP, or roughly connection management, proxying, sniffing & routing, as well as any protocol-specific features. Sub-questions that fall from the main questions are:

- Is the existing functionality in Envoy sufficient to implement new L7 protocols? IE are the current Listeners sufficient for extending to many/most other protocols?
- Are there the right APIs available for developers to implement new protocols?
- How should new protocols be added? It doesn't make sense to bloat the codebase, so is Wasm the right way forward?

At this point I'd love to start the discussion and hear what other people think!

/cc folks who have given me early feedback: @markmandel (from agones) @PiotrSikora @jplevyak @duderino

Assignees
No one assigned

Labels
area/community
design proposal

Projects
None yet

Milestone
No milestone

Linked pull requests
Successfully merging a pull request may close this issue.
None yet

You're receiving notifications because you modified the open/close state.

2 participants

Broadly speaking,
the Cloud Native
landscape is
optimized for HTTP

As a result, projects have assumptions around HTTP

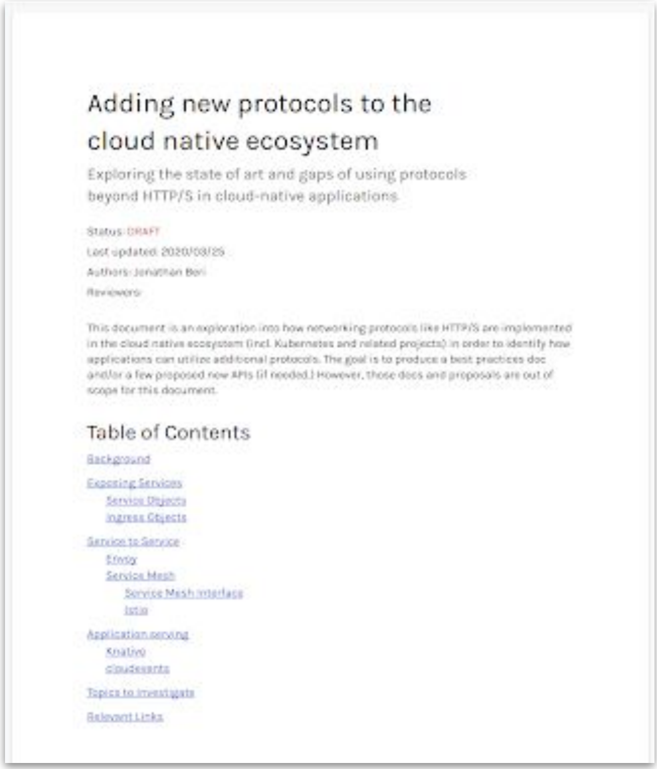
Other domains also use alt. protocols

Agones

Host, Run and Scale dedicated game servers on Kubernetes

Pion

The Open Source, Cross Platform Stack for Real-time Media and Data Communication.



Adding new protocols to the cloud native ecosystem

Exploring the state of art and gaps of using protocols beyond HTTP/S in cloud-native applications

Status: DRAFT

Last updated: 2020/03/25

Authors: Jonathan Beri

Reviewers:

This document is an exploration into how networking protocols like HTTP/S are implemented in the cloud native ecosystem (incl. Kubernetes and related projects) in order to identify how applications can utilize additional protocols. The goal is to produce a best practices doc and/or a few proposed new APIs (if needed.) However, those docs and proposals are out of scope for this document.

Table of Contents

bit.ly/altl7signet

Example: SMI

Service Mesh Interface (SMI) is defining a standard interface for service meshes on Kubernetes and is being developed by multiple vendors. The current focus is HTTP but the goal is to support more protocols in the future. For example, Traffic Specs states, "Each resource in this specification is meant to match 1:1 with a specific protocol. This allows users to define the traffic in a protocol specific fashion." New protocols can be added via a new CRD in the Traffic Specs.

***Takeway**: SMI should be able to support alternative protocols, as it already defines HTTP & TCP.*

Asks

- PTAL! => bit.ly/altl7signet
    - Comments, additional takeaways, new areas to investigate
- Where should I go with this doc?
    - Helpful for current efforts
    - Best way to engage
    - Other SIGs

Summary

- Working on cloud native IoT infrastructure
- Cloud Native is optimized for HTTP
- Alternative protocols hard to implement
- Other domains use alt. protocols
- Making it possible (or easier) to support alt. protocols will help the broader community

These slides @ bit.ly/altl7signet-slides

Thanks!

Jonathan Beri ○ @beriberikix