Namespaces, Security, and Addressing

Andy Reid, Phil Eardley (BT)
Dirk Kutscher (University of Applied Sciences Emden-Leer)
Namespaces, Security, and Addressing

Preview of paper to be presented at SIGCOMM FIRA Workshop
https://conferences.sigcomm.org/sigcomm/2022/workshop-fira.html

Based on work with the Piccolo Project within Celtic-Next
Background to Paper

Rise of container based modularisation of applications

Containerisation simultaneously meets many different requirements including

- Code development (especially to the ‘agile agenda’)
- Service abstraction
- Heterogeneity of development/runtime environments
- Integration (e.g., the modules of a CI/CD pipeline)
- Distribution
- Interaction with networking

Strong case that one solution satisfying all these requirements is a good thing

- But does one size really work for all?

Potential Issues

Logical distribution with in datacentre -> physical distribution

- Edge compute places more constraints than a reasonably homogeneous datacentre infrastructure

Scale of service abstraction is fixed

- Ideally service abstraction should be flexible and recursive

Scale of service abstraction is linked to networking and implementation

- Small modules = flexible implementation but many networking interfaces
- Large modules = small number of networking interfaces but inflexible implementation

“Sidecars” and “proxies” are heavyweight in highly distributed environments

- These link application namespace to network addressing
Two interesting use cases

**Distributed video processing**

Many video applications involve feature extraction and other processing, often based in ML algorithms

**Processing at the camera**
- Increases the cost of the camera
- Make changing the processing algorithms complex

**Processing in the cloud**
- Requires transport of full video bandwidth
- Makes full video stream available to unintended cross-correlations with associated privacy concerns

**Processing at the edge**
- Potentially the best of both

**Automated production facility (smart factory)**

The scale and scope of production automation is becoming more like a networked distributed application

- Very different background and legacy standards

- Greater scale and scope leading to need for modularity in automation design

- Time to re-programme the automation facility to different production applications becoming a key concern
Namespaces and Addressing

Application namespaces

Network addressing

Wide Area Network

Networking interface

Compute infrastructure

IP DA (private or public)

Component A

Component B

Component C

Component D

Sidecar

Virtual switch (shared memory)

Proxy/Load-balancer

Networking interface

Compute infrastructure

IP header 5-tuple (NAT)

IP DA (public)

IP header 5-tuple (NAT)

IP DA (public or private)

Application namespace

REST (http header + IP header)

REST (http header + IP header)
Three areas to be explored

**Bringing the compiler and orchestrator closer together**
- Potentially enables flexible and recursive service abstraction without compromising runtime efficiency
- Avoids “sidecars” and “proxies”

**Defining layering but what is rather than what should be**
- A robust common framework between application and network for service abstraction and transparency

**Primacy of private networking and addressing**
- Future Internet addressing should be based on extensible/contextualisable local addressing in the same way application namespaces work
- Facilitates linkage and can give a basis for security
- This is ‘de facto’ largely the case already