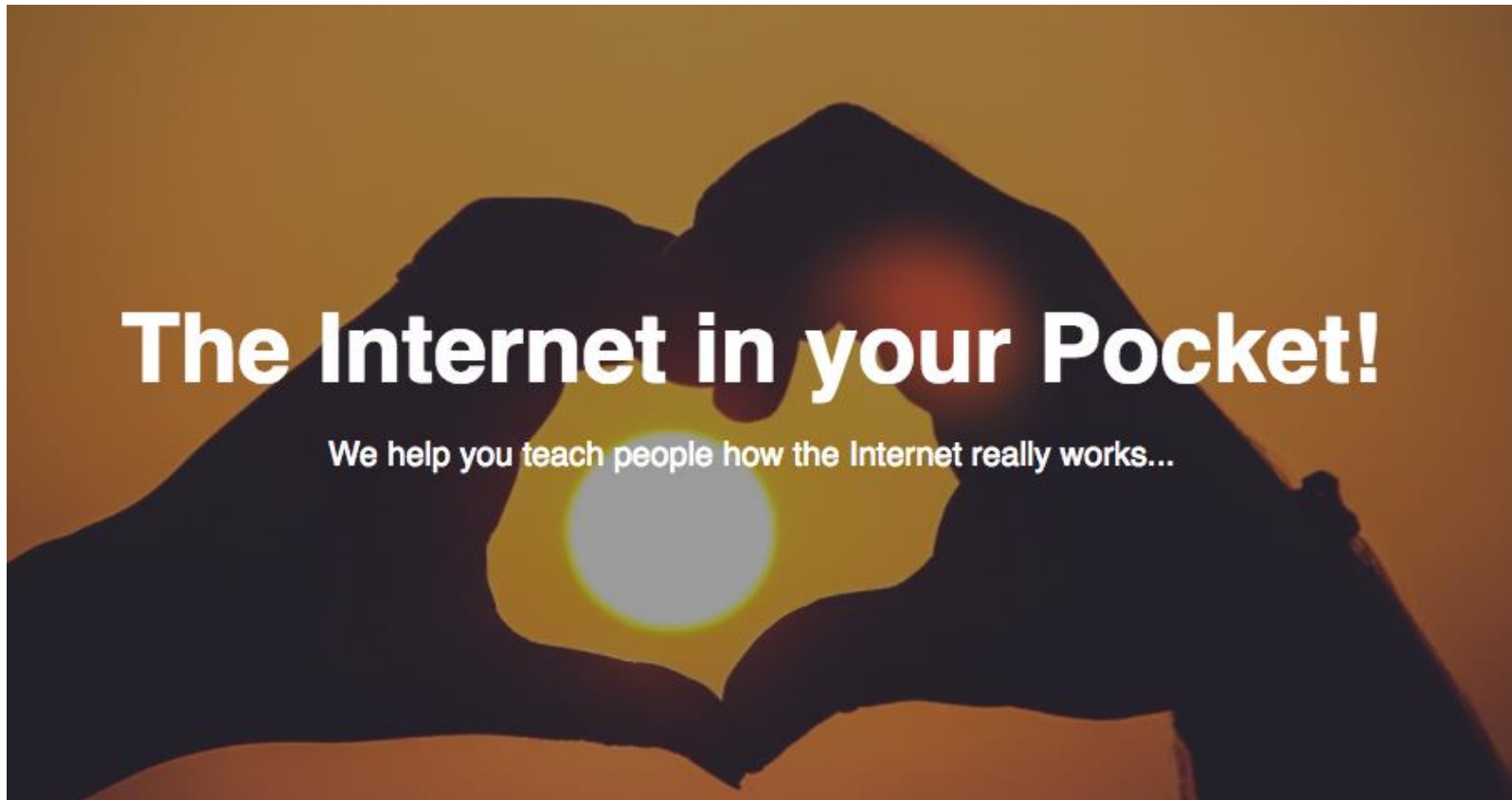


Pocket_Internet



Overall idea

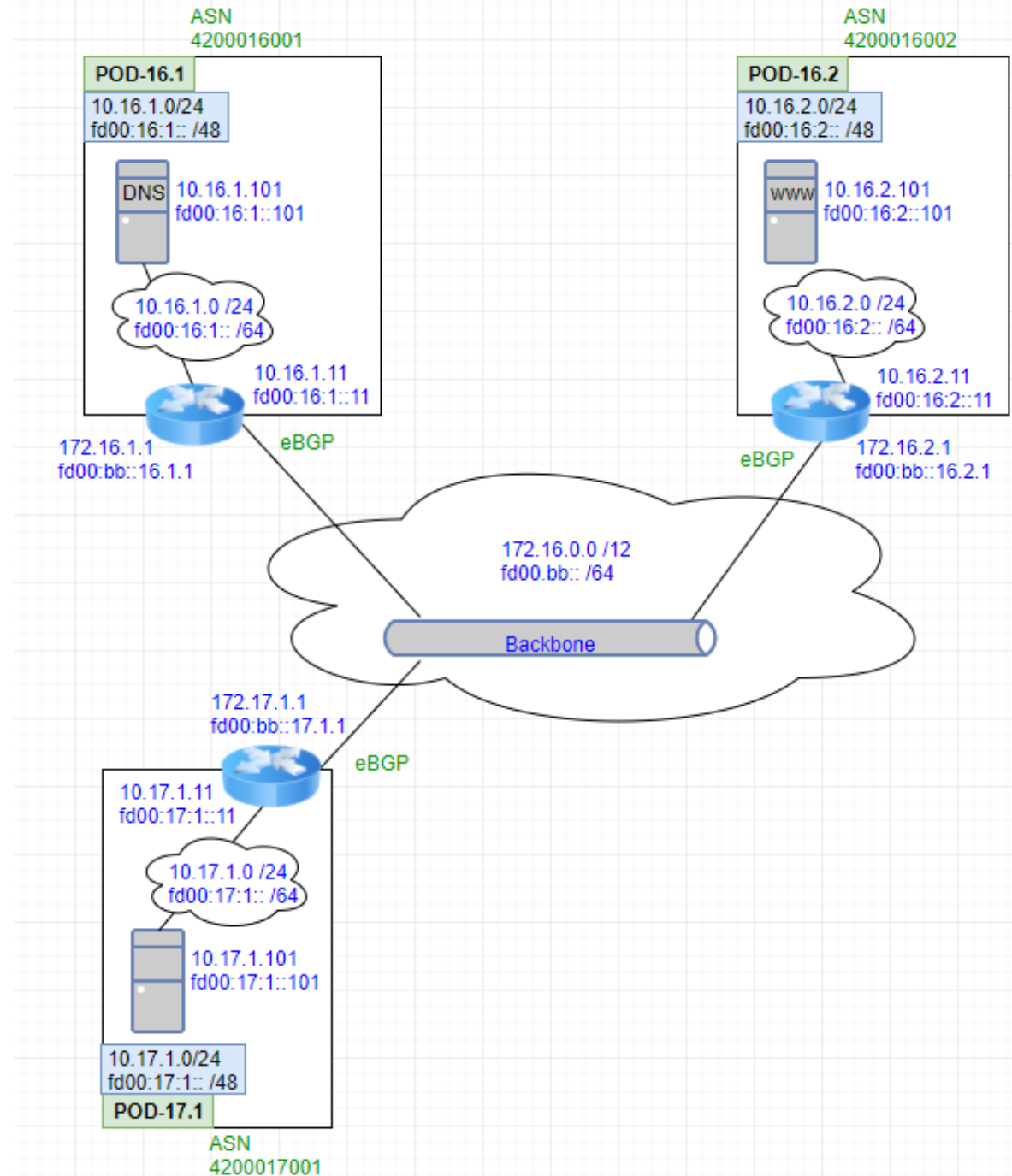
- Create a highly automated lab environment to resemble basic Internet connectivity and services
- Based mostly on a virtualized environment, with the ability to break out to the physical world
- Offer IPv4, IPv4+IPv6 and IPv6-only environments, so trainees understand and can experience the difference
- Based on a modular architecture so that different lab environments can be set up

Fundamentals

- Docker + automation is used to create repeatable lab setups
- Create a single backbone for interconnects
- Define modular PODs of different type to connect to this backbone
- POD to POD communication is via BGP routing (defines the topology)

Simple example:

- Three PODs:
 - One with a DNS server
 - One with a http server
 - One with a client
- All PODs with just a single backbone router
- Dual-stack setup with IPv4 + IPv6



Addressing + DNS names

- Each interface (for routers & server instances) can have IPv4, IPv6 or both configured
- DNS names are pre-configured for every interface, one for dual-stack (resolving as AAAA + A) and one each for v4 and v6 (resolving as A, or AAAA only)
- Naming is specific per POD, with an addtl. global domain for cross-POD testing

Container instances

- Defined base container image, including network tools (traceroute, curl,...)
Every container is based on that.
- For the initial setup:
 - Routing containers run BIRD
[due to BIRDs restrictions, there is one container for BIRD4 and one for BIRD6]
 - DNS container runs BIND
 - www container runs NGINX
 - Client container offers SSH



Build-process

- Proof-of-concept builds with docker-compose through a yaml file
 - Creates the backbone and POD networks
 - Instantiates all containers and connects them to the right networks
- Started work on defining the lab setup through a config file, which specifies the logic of a lab:
 - How many and which kind of PODs
 - Specify peering relationships
 - Specify protocol (dual-stack, IPv4, IPv6-only)

Use case examples

- Simple dual-stack BGP lab, checking out BIRD and routing
- Advanced routing workshops with filters, route policies, ...
- IXP, how does an internet exchange work
- DNS, how name resolution enables migration to IPv6

Evolution

- Define more complex PODs, such as simulating an ISP
- Define IPv6 transition PODs, such as one for NAT64
- Define an exit-node, which can link the virtual environment out to a physical network, so that real hardware can be also used for testing, such as client OSes, network hardware, ...
- Still need more automation, i.e. to create DNS zone files.
- Create a front-end for ease of lab setup

Not the end of the story...

;-)

Contributors:



Cristian Sirbu

Evangelos Balaskas

Henrik Kramshøj

Harry Reeder

Samer Lahoud

Andy Mindnich