EduCDN

Distributed Systems Challenge Task
FS2025

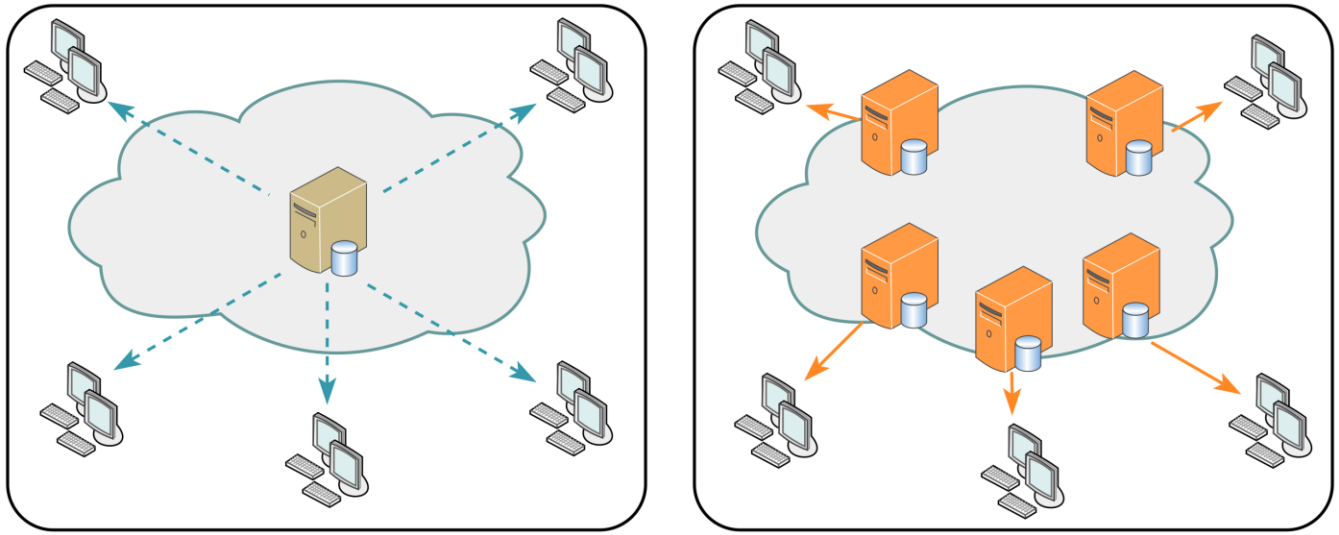
Ramon Bister and Roman Weber

26.05.2025


What is a CDN?
Why did we build one?

Definition

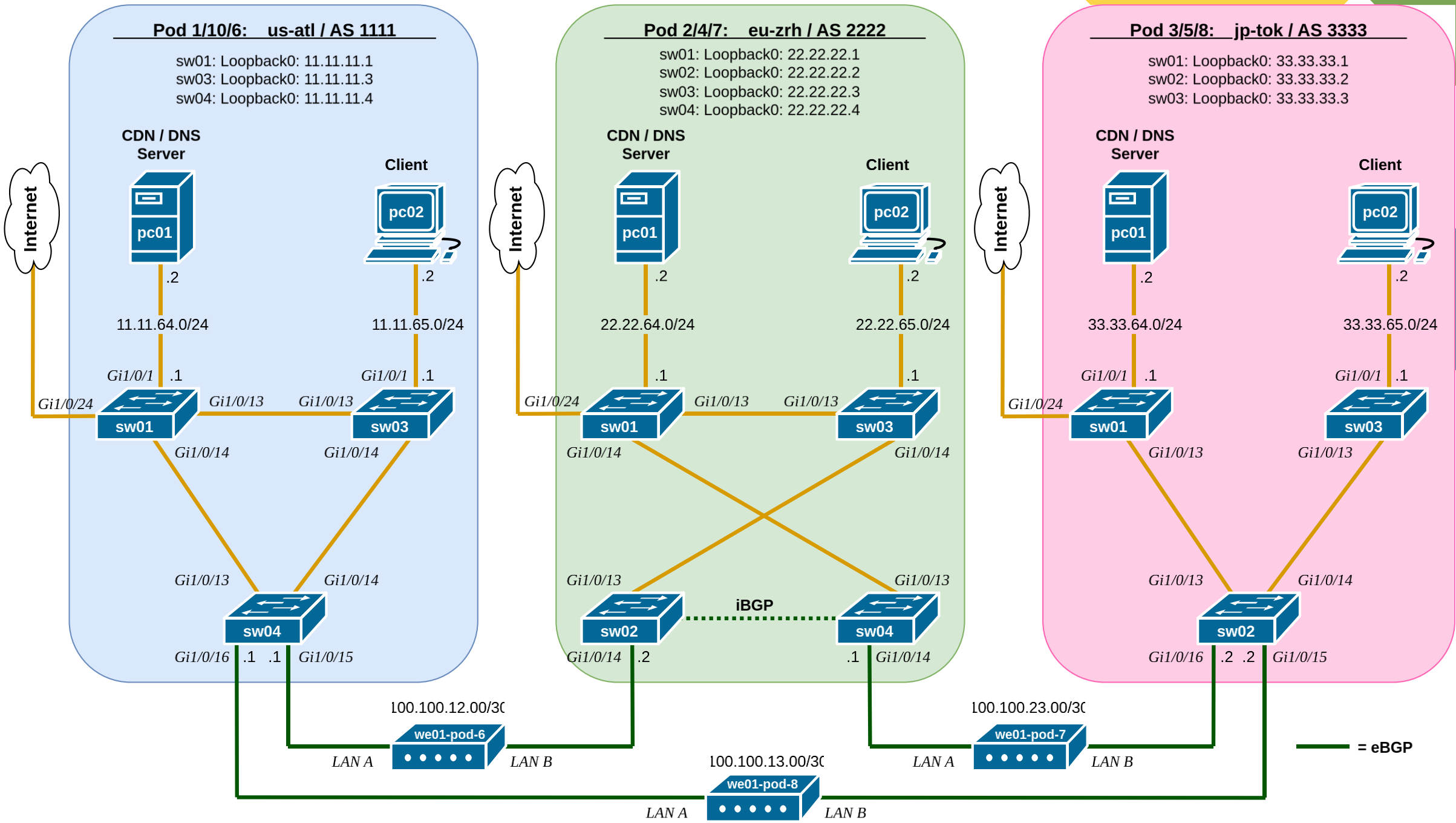
A **content delivery network** or **content distribution network** (**CDN**) is a geographically distributed network of proxy servers and their data centers. The goal is to provide high availability and performance ("speed") by distributing the service spatially relative to end users.



Source: https://en.wikipedia.org/wiki/Content_delivery_network

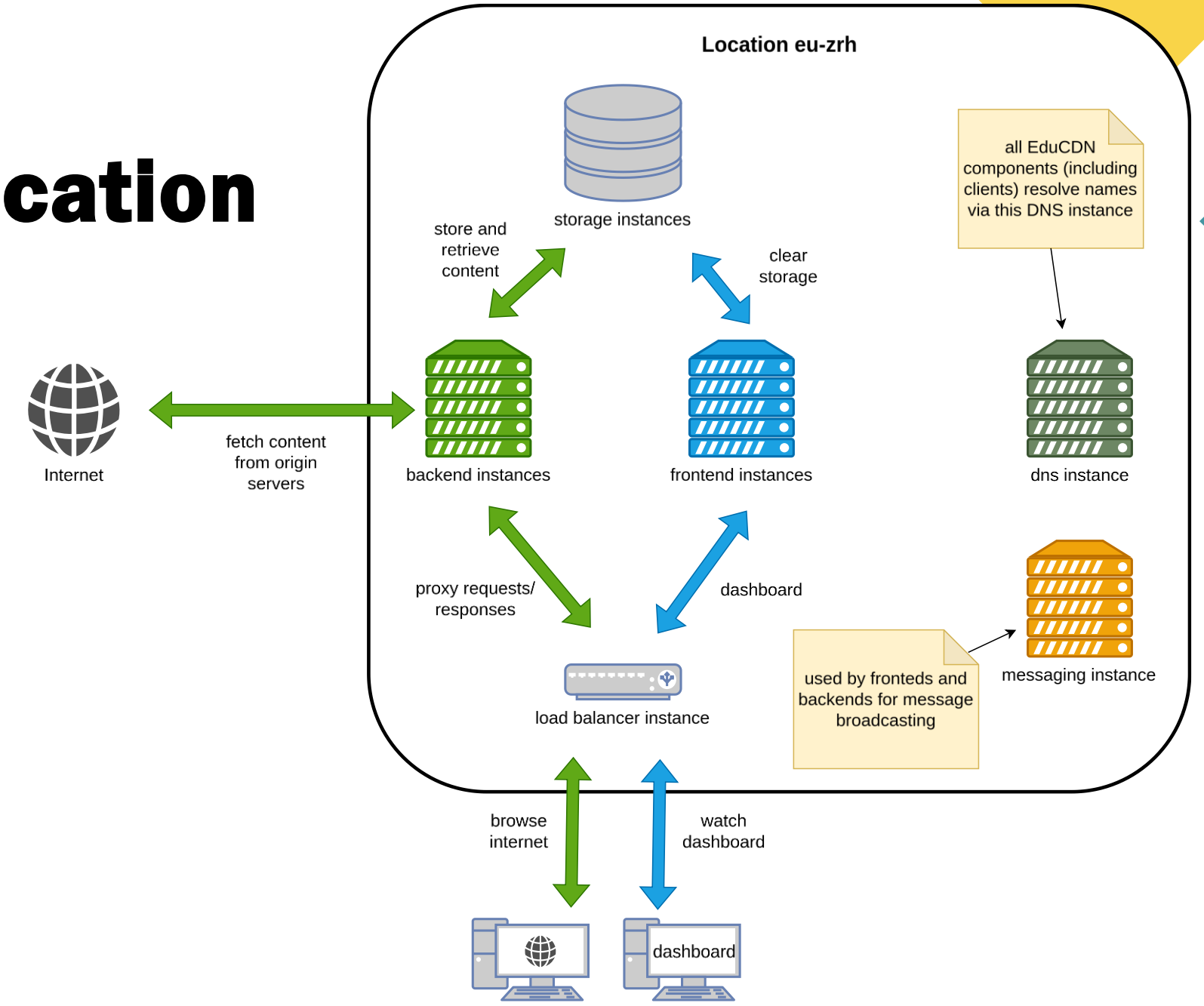
Why did we build EduCDN?

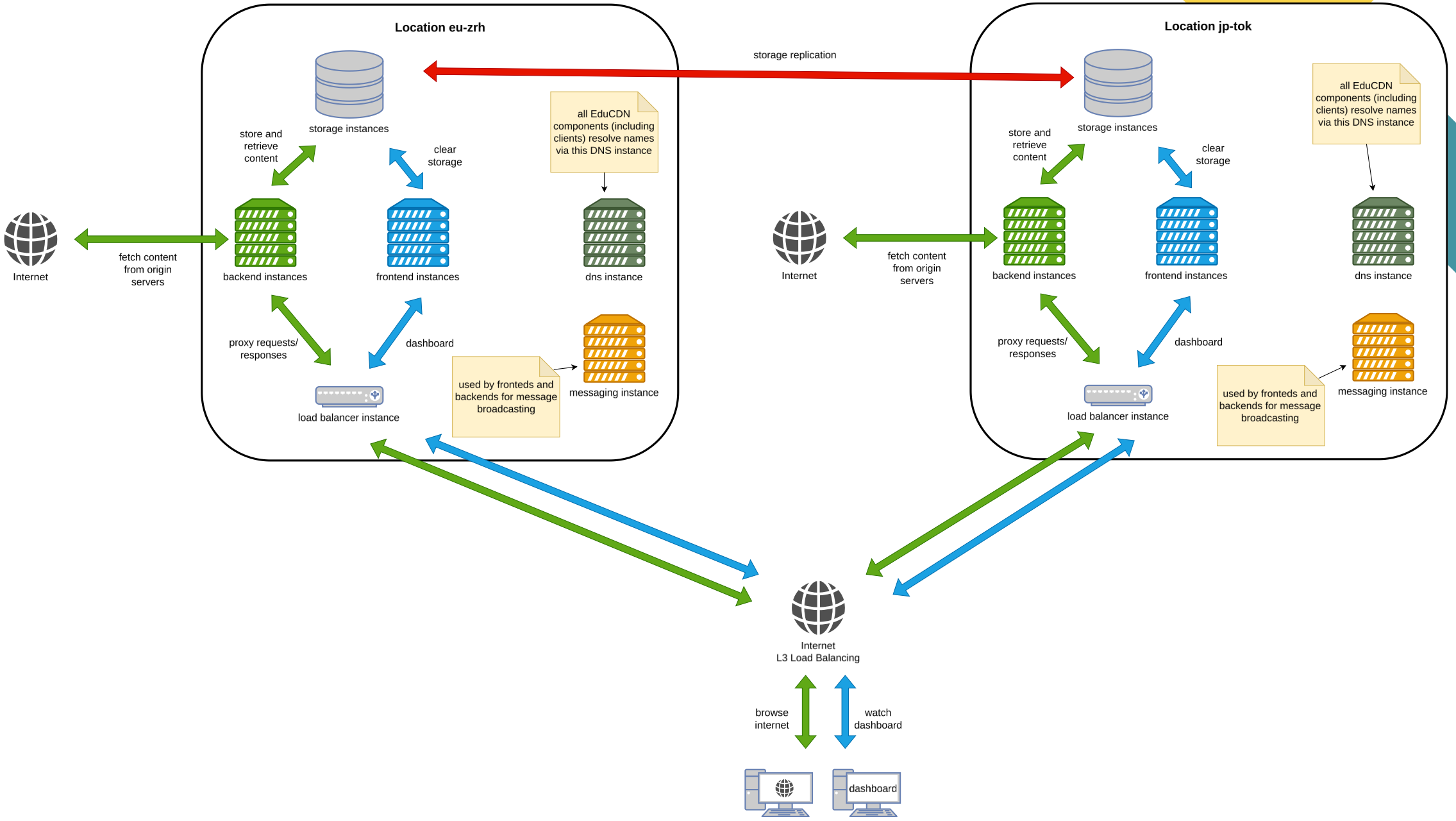
- The institute for networks and security (INS) requires an **education purpose CDN** because:
  - The CDN topic was added to the curriculum of the CN2 module in FS2025
  - Students should get practical experience with the concepts by:
    - Configuring L3 load balancing with BGP anycast
    - Seeing HTTP caching in action
    - Seeing how the CDN does its job based on events displayed on the dashboard


Pod 1/10/6: us-atl / AS 1111
sw01: Loopback0: 11.11.11.1
sw03: Loopback0: 11.11.11.3
sw04: Loopback0: 11.11.11.4
Pod 2/4/7: eu-zrh / AS 2222
sw01: Loopback0: 22.22.22.1
sw02: Loopback0: 22.22.22.2
sw03: Loopback0: 22.22.22.3
sw04: Loopback0: 22.22.22.4
Pod 3/5/8: jp-tok / AS 3333
sw01: Loopback0: 33.33.33.1
sw02: Loopback0: 33.33.33.2
sw03: Loopback0: 33.33.33.3
CDN / DNS Server
Client
Internet
pc01
pc02
sw01
sw02
sw03
sw04
.2
11.11.64.0/24
11.11.65.0/24
22.22.64.0/24
22.22.65.0/24
33.33.64.0/24
33.33.65.0/24
Gi1/0/1
.1
Gi1/0/24
Gi1/0/13
Gi1/0/14
Gi1/0/15
Gi1/0/16
iBGP
100.100.12.00/30
100.100.23.00/30
100.100.13.00/30
we01-pod-6
we01-pod-7
we01-pod-8
LAN A
LAN B
= eBGP


Architecture

Single Location




Location eu-zrh
storage instances
store and retrieve content
clear storage
all EduCDN components (including clients) resolve names via this DNS instance
Internet
fetch content from origin servers
backend instances
frontend instances
dns instance
proxy requests/ responses
dashboard
used by fronteds and backends for message broadcasting
messaging instance
load balancer instance
storage replication
Location jp-tok
storage instances
store and retrieve content
clear storage
all EduCDN components (including clients) resolve names via this DNS instance
Internet
fetch content from origin servers
backend instances
frontend instances
dns instance
proxy requests/ responses
dashboard
used by fronteds and backends for message broadcasting
messaging instance
load balancer instance
Internet
L3 Load Balancing
browse internet
watch dashboard
dashboard

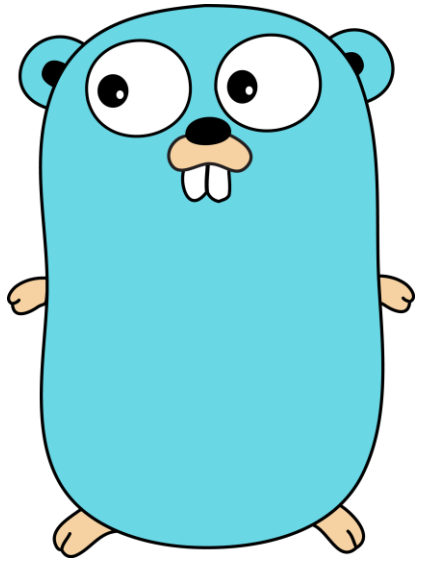

Components

Application Layer Load Balancer

https://github.com/traefik/traefik

- Load balances HTTP requests to
  - Backend Instances (.* with lowest priority)
  - Frontend Instances (**dashboard.educdn.net**)
  - Storage Instances (**storage.educdn.net**)
- Uses self-signed TLS certificate valid for
  - All **CDN-enabled** websites
  - **dashboard.educdn.net and storage.educdn.net**

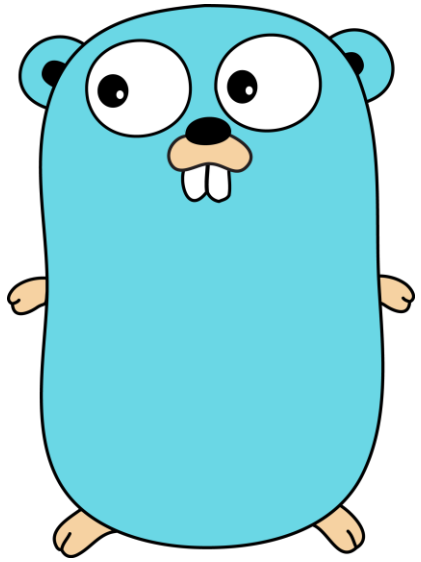
Backend Instances



- Act as proxy server with the capability to
  - In case of **cache miss:**
    - Fetch content from origin web server
    - Write fetched content including required metadata to cache and client
  - In case of **cache hit:**
    - Retrieve content and metadata from cache
    - Write cached response to client

https://gin-gonic.com/

Frontend Instances



- Serve dashboard with Go Server Side Rendering (SSR)
- Buffer events from message queue
- Broadcast events to all connected web browsers with Server Sent Events (SSE)

https://gin-gonic.com/

Storage Instances (MinIO)

- Self-replicating S3 object storage to
  - Replicate cache among locations
  - Store HTTP response bodies and HTTP header data

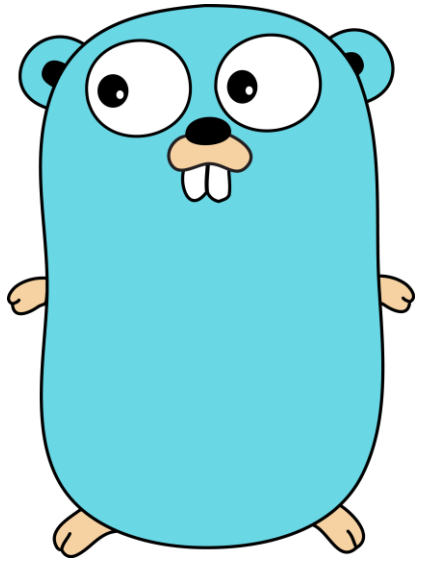
https://github.com/minio/minio

Messaging Instances (NSQ)

- Realtime distributed messaging platform
  - Backends of any location produce messages
  - Frontends of any location consume messages
- Solves the problem that a user connected to an **arbitrary frontend at an arbitrary location** can receive all messages sent by an **arbitrary backend at an arbitrary location**
- **No need for central state instead messages are broadcasted.**

https://github.com/nsqio/nsq

DNS Instances



- Based on https://github.com/miekg/dns
- Resolves EduCDN services and CDN-enabled websites to either the unicast or anycast address of the location (depending on configuration)
- Other DNS queries are forwarded to the configured external server to get the **actual IP address**
- Has **split-horizon** DNS functionality
  - **Queries from backend instances are always forwarded** to the configured external server
- Discovers backend instances automatically in configurable interval

Client-Side Frontend

- Follows the model, view controller (MVC) architecture
- Connects to *Go Frontend Instance* to subscribe to server sent events
- Displays the server sent events in HTML tables with filter possibilities
- Allows to the user to trigger storage/cache deletion


Design Decisions

Single Binary

```
An education purpose content delivery network with a proxy system as backend connected
to a self-replicating storage, a fronted that shows state information, an integrated
DNS service and a configuration bootstrapper.

Usage:
  educdn [command]

Available Commands:
  backend     Start the educdn backend
  bootstrap   Bootstrap the configuration files used for the deployment
  dns         Start the DNS service
  frontend    Start the educdn frontend
  help        Help about any command

Flags:
      --config string   The config file to be used
  -h, --help            help for educdn
  -v, --version         version for educdn
```

Single Source of Truth for Bootstrap

```
global:
  anycast:
    enabled: false
    address: 7.7.7.7
  enabledDomains:
    - www.mfv-thal.ch
    - nzz.ch
    - \*.nzz.ch # match all subdomains of nzz.ch (e.g www.nzz.ch / img.nzz.ch)
    - \*.static-nzz.ch
    - bitmovin.com
    - \*.bitmovin.com
  <…>
local:
  <…>
  backend:
    replicas: 1
    address: 0.0.0.0
    port: 8080
 <…>
```

Other Important Decisions

- Storage Replication based on MinIO
- Integrated Domain Name System
- Public Key Infrastructure
  - Own CA
  - Certificates for each enabled domain explictly trusted by clients
- Distributed Messaging System to avoid centralized state


Demonstration

Demo overview