

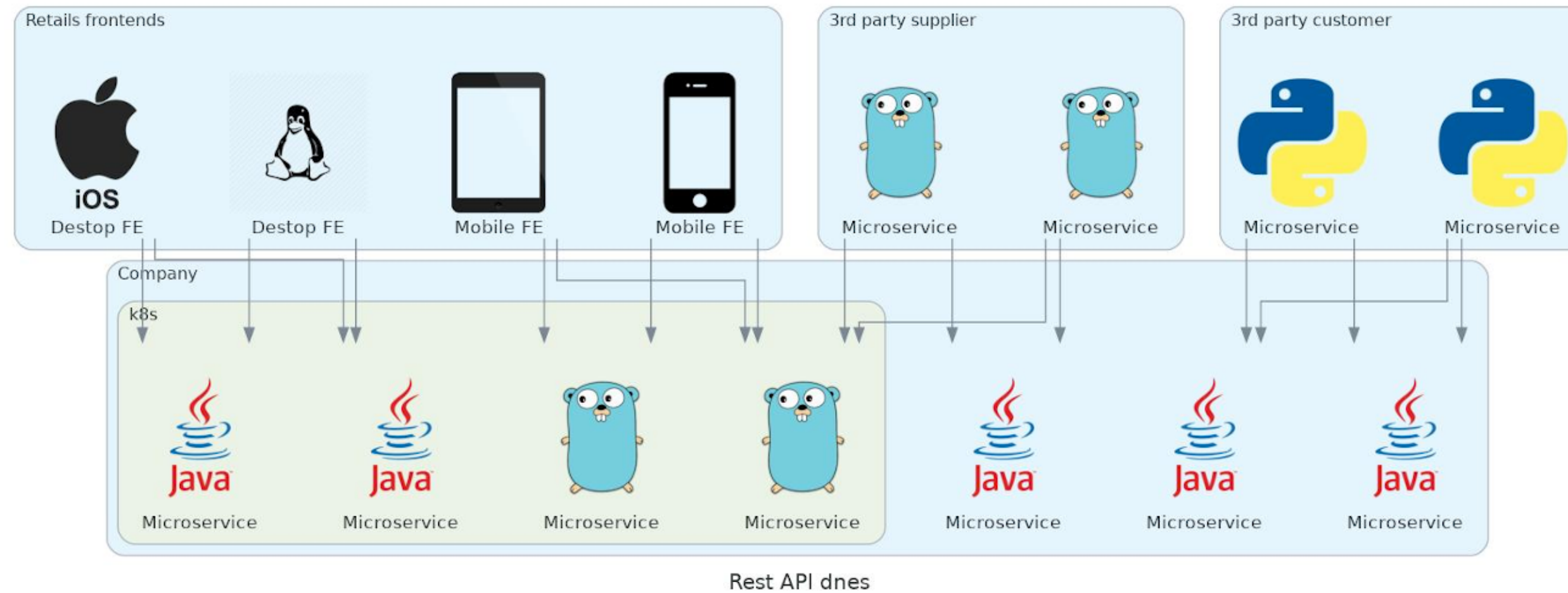
Make ~~your~~ api great again!

Co API Gateway je a proč ji potřebujeme

JAN.DVORAK@VITRACE.IO



Make your api great again!



JAN.DVORAK@VITRACE.IO



Make ~~your~~ api great again!

REST API JE DNES PŘEDEVŠÍM ZDROJEM DAT!

JAN.DVORAK@VITRACE.IO



Make ~~your~~ api great again!

DATA POTŘEBUJEME

- konsolidovat
- bezpečně publikovat
- integrovat

...zkrátka mít nad nimi kontrolu!

JAN.DVORAK@VITRACE.IO



Make ~~your~~ api great again!

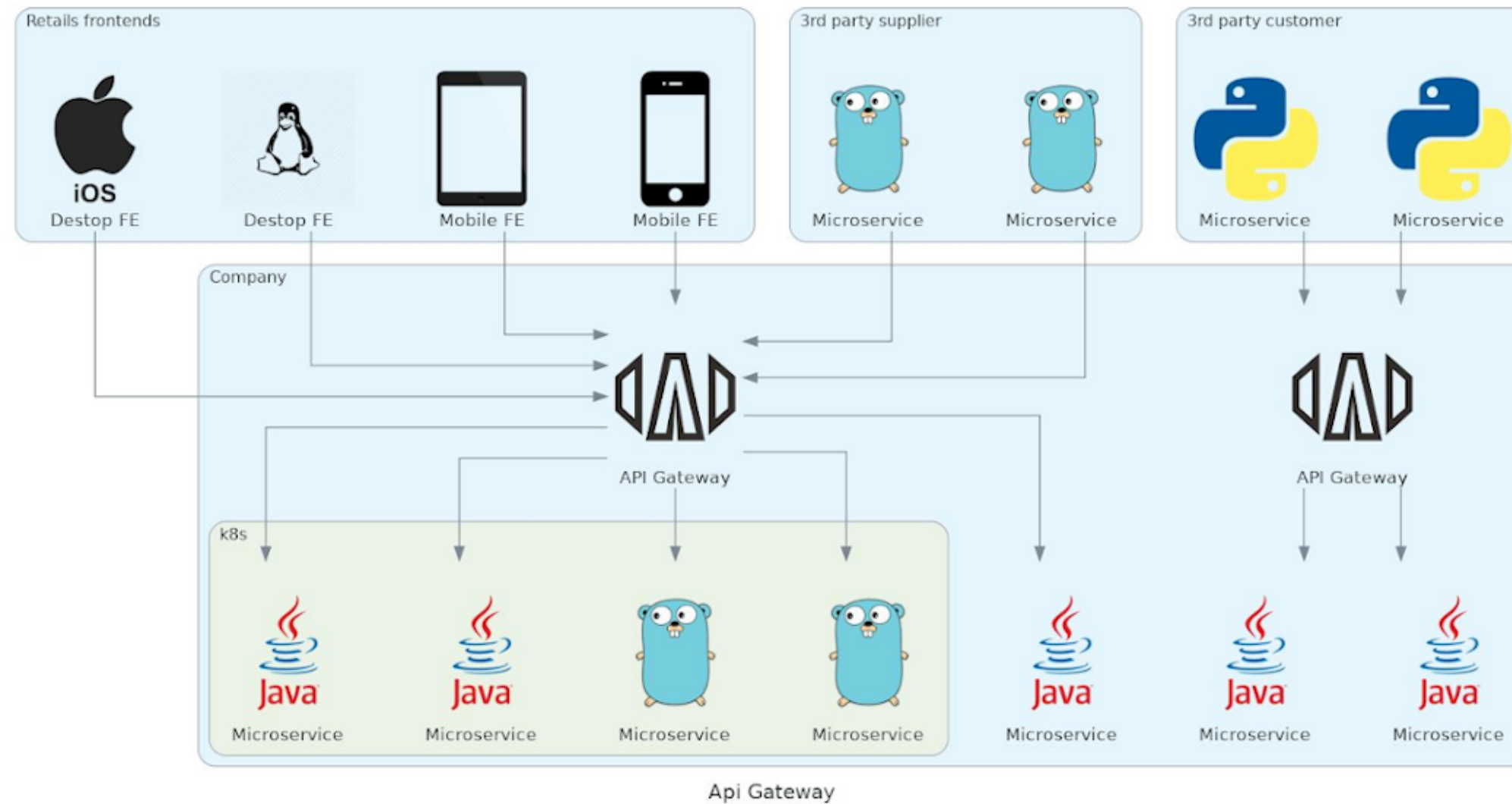
CO JE API GATEWAY

- Reverzní proxy která:
 - je orientována na práci s REST požadavky
 - umožňuje jejich filtering, transformace, apod.
 - integruje naše API s dalšími prvky naší infrastruktury.
 - další možné funkce:
 - service discovery
 - rate limiting
 - circuit breaking atd

JAN.DVORAK@VITRACE.IO



Make ~~your~~ api great again!



JAN.DVORAK@VITRACE.IO



Make ~~your~~ api great again!

Api Gateway

Zaměřena na manipulaci s rest požadavky

Mix obecné a "fine grained" konfigurace

Transformace request/response, views..

Service owner: vývoj/infrastruktura

Reverse Proxy/Ingress

Zaměření na přístup k aplikaci

Obecná konfigurace

SSL terminace, segmentace

Service owner: infrastruktura

JAN.DVORAK@VITRACE.IO



Make ~~y~~our api great again!

DYI Api Gateway

JAN.DVORAK@VITRACE.IO



Make ~~your~~ api great again!



JAN.DVORAK@VITRACE.IO



Make ~~y~~our api great again!

OPENRESTY (NGINX + LUA VM) - API GATEWAY PRO ODVÁŽNÉ :)

- Modifikování flow pomocí lua skriptů
- Spousta pluginů
- Často jako základ jiných sw (Kong, různá CMS apod)
- Jedná se spíše o framework

<https://openresty.org/>
<https://github.com/openresty/lua-nginx-module>
<https://openresty-reference.readthedocs.io/en/latest/Directives/>

JAN.DVORAK@VITRACE.IO



Make ~~your~~ api great again!

OPENRESTY - PŘÍKLAD KONFIGURACE

```
location /mutex {
    set $locid 14;
    access_by_lua_block {

        -- Main
        local mutexes = ngx.shared.mutexes;
        local session_id = ngx.var.arg_id;
        local mutex, err = mutexes:get(session_id)

        if mutex == nil then
            local ok, err, force = mutexes:add(session_id,1, 5);
            ngx.exit(ngx.HTTP_CREATED);
        end;

        ngx.exit(ngx.HTTP_NO_CONTENT);
    }
}
```

JAN.DVORAK@VITRACE.IO



Make ~~your~~ api great again!

PROČ OPENRESTY

- Je to Nginx :)
- Lua skriptování je jednoduchý ale velmi mocný nástroj
- Spousta pluginů
- Široká komunita
- Slušný výkon

JAN.DVORAK@VITRACE.IO



Make ~~your~~ api great again!

OPENRESTY - TEMNÁ STRANA SÍLY

- Mix infrastruktury a vývoje (nedá se tomu uplně vyhnout :)
- Naročné na údržbu a deployment
- Některé pluginy jsou oneman show s diskutabilní podporou
- Občas složité kódování logiky (např. závislé na response body)
 - někdy může být vhodnější NGINX Javascript (je možné oba moduly kombinovat)

JAN.DVORAK@VITRACE.IO



Make ~~your~~ api great again!

KRAKEND

- CE je Stateless
- Napsán v golangu
- Vynikající dokumentace
- Konfigurace je v jsonu, možnost použití designeru
- Podpora pro: Filtering a transformace http requestů
 - OIDC
 - Rate Limiting
 - Circuit breaking
- Rozšiřitelný pomocí custom pluginů

<https://www.krakend.io/>
<https://www.krakend.io/docs/overview/introduction/>
<https://designer.krakend.io/>

JAN.DVORAK@VITRACE.IO



Make your api great again!

KRAKEND - PŘÍKLAD KONFIGURACE

```
{
  "endpoint": "/simple/articles",
  "method": "GET",
  "extra_config": {
    "github.com/devopsfaith/krakend-jose/validator": {
      "alg": "RS256",
      "roles_key": "groups",
      "roles": ["simple-admin", "simple-user"],
      "jwk_url": "https://secure-keycloak-keycloak.route.local/auth/realms/lab/protocol/openid-connect/certs",
      "disable_jwk_security": true
    }
  },
  "backend": [
    {
      "url_pattern": "/articles",
      "encoding": "json",
      "is_collection": true,
      "extra_config": {
        "github.com/devopsfaith/krakend/proxy": {
          "flatmap_filter": [
```

JAN.DVORAK@VITRACE.IO



Make ~~your~~ api great again!

DALŠÍ PROJEKTY

- 3scale
- Kong
- Tyk
- Traefik
- Plus projekty v rámci public cloud subscripce

JAN.DVORAK@VITRACE.IO



Make ~~your~~ api great again!

PASTI, PASTI, PASTICKY

- Bude naše apigw statefull nebo stateless ?
 - specialně pozor na nasazení v k8s
 - statefull - možný dopad na škálování
- V kubernetes některé funkce zvolené (dodané) CNI - pozor na to.
- Jak nasazení ovlivní provoz a deployment backend služeb ?
- Jedna nebo více instancí ?
- Chceme některé funkcionality přenést z aplikace na apigw ?
 - typicky autorizace, session management apod...
- Máme dostatek lidských zdrojů na implementaci požadovaného řešení ?

JAN.DVORAK@VITRACE.IO



Make ~~your~~ api great again!

CO POMŮŽE

- V maximální míře využívat OpenAPI
- generování konfigurace na základě swaggeru
- Rozděl a panuj!
- Méně je více

JAN.DVORAK@VITRACE.IO



Make ~~your~~ api great again!



QUESTIONS ?

JAN.DVORAK@VITRACE.IO



