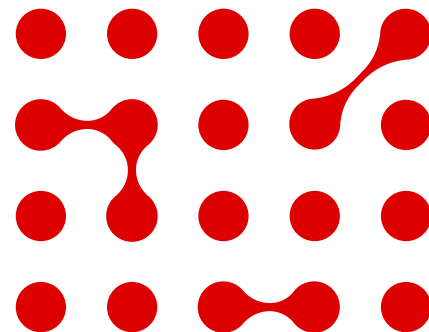
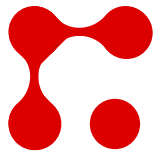


# ISLETNET

PETR BĚLOHLÁVEK

ITERAIT ŠKOLENÍ PRO IKEM 22. 9. 2021





# OBSAH



## ITERAIT

Kontaktní osoby

01

03

## ARCHITEKTURA APLIKACE

Frontend, backend,  
storage, shepherd, pruner

## ISLETNET

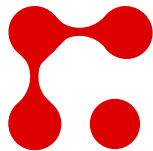
Development workflow,  
aplikace, research

02

04

## DEPLOYMENT

Kontejnery,  
docker-compose, tagy,  
jak spustit/ vypnout/  
aktualizovat



# OBSAH



## MONITORING

Dummy tasks, docker  
healthcheck

05

07

## BUDOUCÍ VÝVOJ

Funkční a tech  
požadavky, research,  
deops/deploy/status

## TROUBLESHOOTING

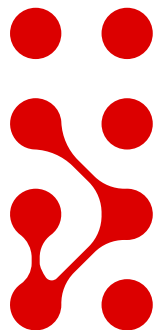
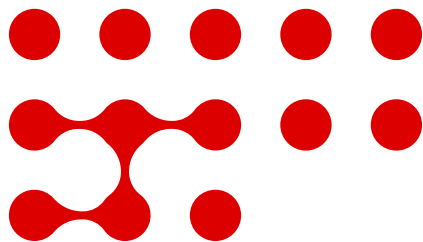
Jak restartovat, pročistit  
data, use cases

06

08

## SHRNUTÍ

Závěr a diskuze



# 01

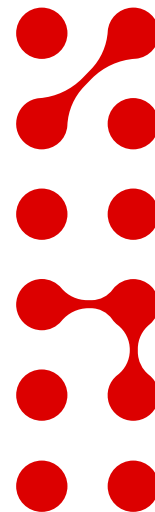
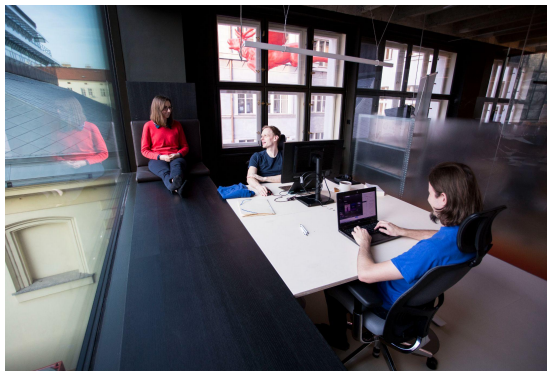
## ITERAIT

- Machine learning a umělá intelligence

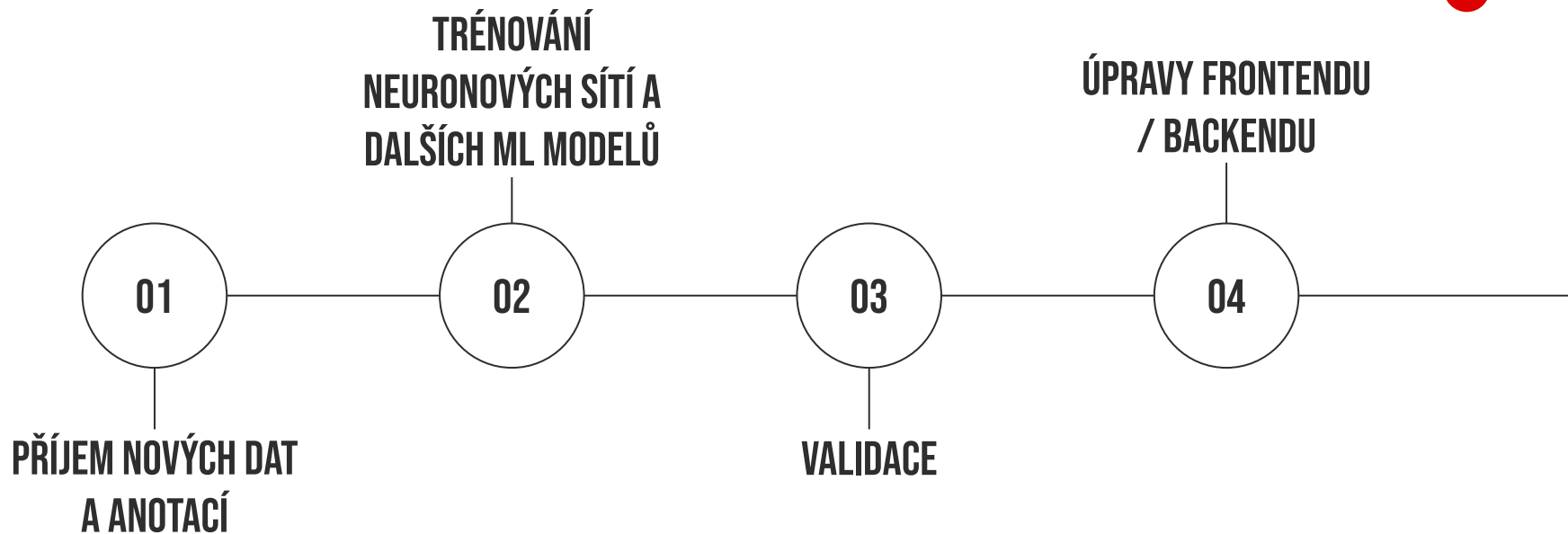
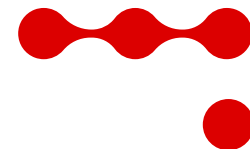
# ITERAIT

Jsme česká technologická společnost se sídlem v Praze. Zaměřujeme se na vývoj umělé inteligence a její využití v různých oblastech.

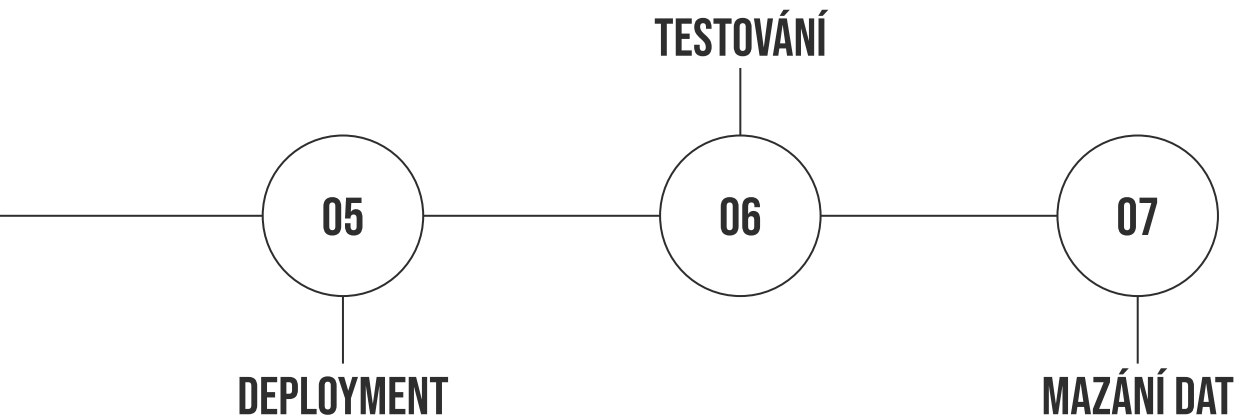
Kontaktní osoba:  
**Petr Bělohlávek**  
+420 731 012 334  
[petr.belohlavek@iterait.com](mailto:petr.belohlavek@iterait.com)



# WORKFLOW



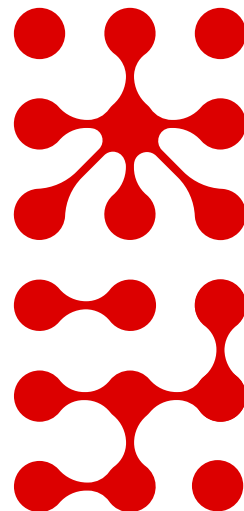
# WORKFLOW

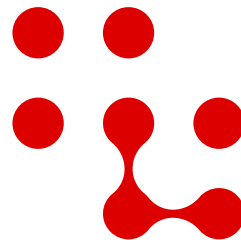




## ISLETNET

Zrychlení transplantace ■  
Langerhansových ostrůvků





## APLIKAČNÍ ČÁST



- Webová aplikace
- UI pro uživatele
- Produkční neuronové sítě



## VÝZKUMNÁ ČÁST



- přípravné algoritmy pro vědecké publikace
- vizualizace
- automatizované zpracování dat
- trénování ML modelů

# ISLETNET - APLIKAČNÍ ČÁST

IKEM

Current version: 2020-01-29  
Last network training: 2019-03-23

Home

Clinical Islet Isolation

Validation Tools

Simple Comparison

Pixel-to-Pixel Comparison

My Segmentations

Edit batch

1

Remove batch

Image navigation

islet.jpg

Microscope: 1  
Camera: 1  
Objective: 1x  
Pixel size (μm/px): 2.600  
Minimum islet size (μm): 50  
Sample dilution: 1  
Islets: 27  
IE (table): 30  
IE (ellipse): 26  
% (purity, area): 15  
mL (pellet): 1.0

Include

Exclude

Use for training

Contours

Ellipses

Original

IsletIDs

Exocrine tissue

Islets + Exo

Islets

Histogram

Batch summary (mean, n = 1/1):

27 islets

30 IE (table)

26 IE (ellipse)

15 % (purity)

islet.jpg

27

29.97

25.64

15

Add another batch

Show summary

10

IKEM

# ISLETNET - VÝZKUMNÁ ČÁST

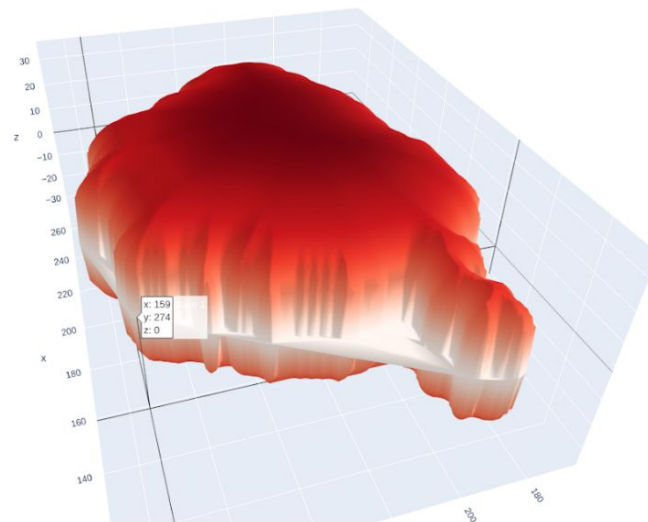


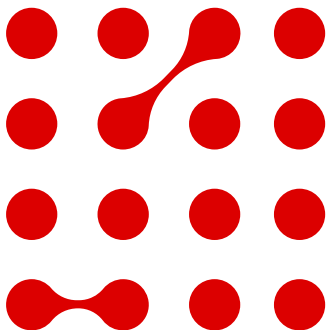
Jednorázové skripty potřebné pro přípravu vědeckých studií

- vizualizace chyb ML modelů
- počítání korelací výsledků s “ground truth”
- příprava datových souborů pro následnou analýzu
- interaktivní 3D vizualizace ostrůvků
- kontrola běhu aplikace

Trénování ML modelů

- načítání dat a anotací
- pipeline pro augmentaci dat ([hipipe](#))
- trénování samotných neuronových sítí ([emloop](#))

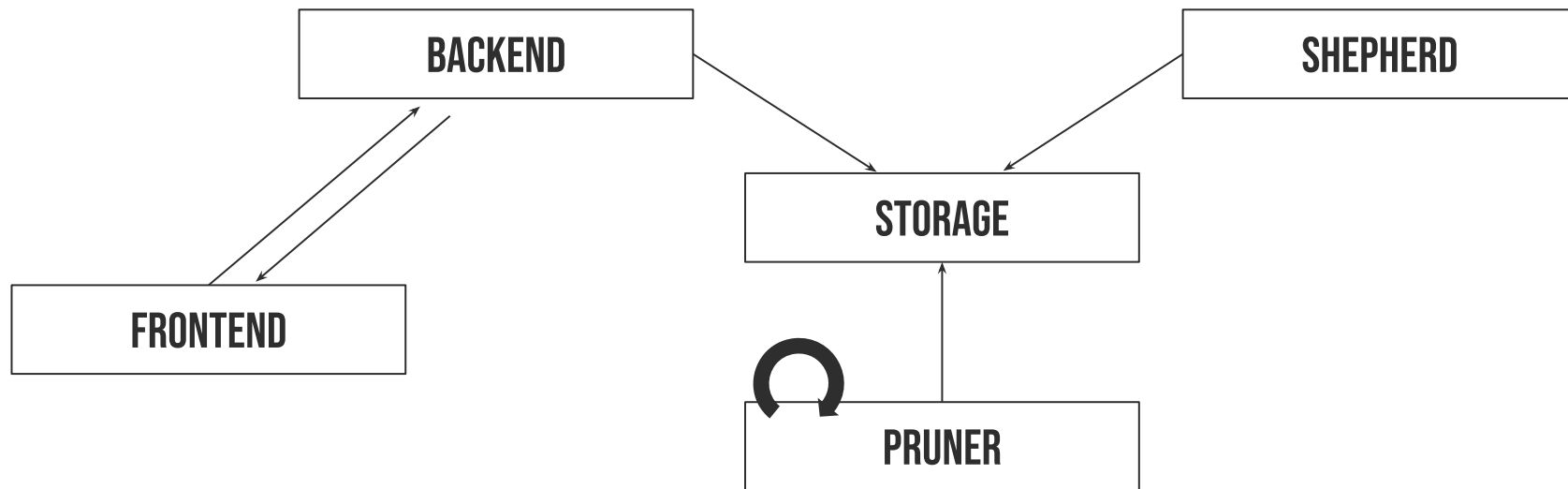
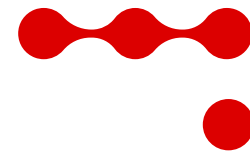




## ARCHITEKTURA APLIKACE

- Frontend, backend,  
shepherd, storage, pruner

# ARCHITEKTURA APLIKAČNÍ ČÁSTI



# ARCHITEKTURA - FRONTEND



UI/UX pro uživatele

Single-page aplikace

Asynchronní komunikace s backendem (REST)

Framework React

- Komponentový systém
- Dependency injection
- Maximum stateless komponent
- Jeden centrální stav aplikace (obrázky, request id, default values)

Typescript

# ARCHITEKTURA - BACKEND



Python - Flask server

REST API

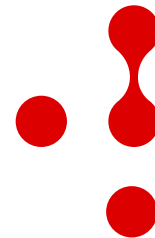
- možno připojit jakýkoliv frontend, mobilní aplikaci (např. Adam Koza) apod
- OpenAPI specifikace

Příjem požadavků z frontendu

Registrace nových jobů v shepherdovi

Zpracování výsledků

- agregační statistiky
- pdf reporty, tabulky, archivy
- mailing
- tvorba datových souborů ke stažení



# ARCHITEKTURA - SHEPHERD



Asynchronní worker + task manager

Přijímá požadavky o nových výpočtech a jejich umístění ve storage

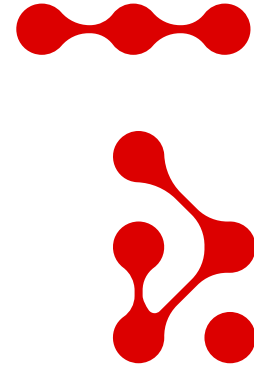
Více nezávislých GPU

Střídání několika běžících neuronových sítí

Sekvenčně analyzuje požadované snímky

- Preprocessing (CPU)
- Neuronová síť (GPU)
- Postprocessing (CPU)

# ARCHITEKTURA - SHEPHERD



1. Preprocessing
  - Validace dat
  - Zpracování snímků (scaling, barevnost masky, ...)
2. Neuronová síť
  - End-to-end autoencoder
  - Klasifikace pixelů do tří kategorií: ostrůvek, exo. tkáň, pozadí
  - TensorFlow v1, běží na GPU
3. Postprocessing
  - Nalezení ostrůvků
  - Počítání jejich objemů dle jednotlivých metod
  - Detekce fragmentace a dalších statistik
  - Tvorba kontur apod.

# ARCHITEKTURA - STORAGE



Slouží jako sdílené úložiště pro backend a shepherd

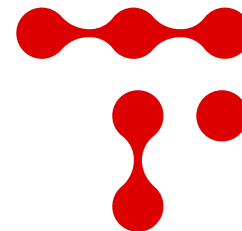
- input data: požadavky, snímky a meta
- output data: výsledky, vizualizace, statistiky

Kompatibilní s cloudovými úložišti S3

- AWS, Azure, ..
- Škálovatelné do budoucna

Minio

# ARCHITEKTURA - PRUNER



Data se smí udržovat jen po určitou dobu

Periodicky mažeme stará data

- snímky
- výsledky

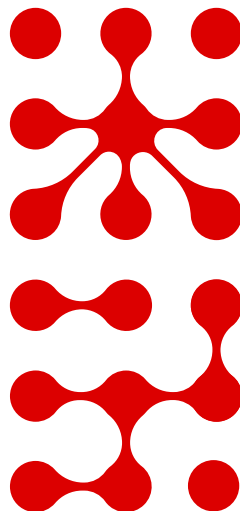
Uživatel může označit některé svoje snímky pro použití v tréninku

- nemažou se nikdy
- jednorázové stažení při dalším tréninku

# 04

## DEPLOYMENT

Docker-compose, verzování ■  
a ovládání



# DEPLOYMENT



Jednotlivé části jsou zkompilevané do separátních docker images

- base image většinou Ubuntu nebo Arch linux
- storage minio/minio
- obrazy jsou uloženy v docker registry [docker.iterait.com](https://docker.iterait.com) (on demand)
- obrazy stahujeme my, příp je možnost dohodnout se na přístupu do registry
- Shepherd vyžaduje podporu GPU (tzn funkční CUDA, CuDNN, nvidia-smi, ...)
  - může běžet i na CPU, ale je velmi pomalý

Pro každou část stačí jeden kontejner

- v případě škálování je možné duplikovat shepherd

# DOCKER DAEMON



Runtime: nvidia default runtime

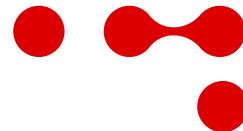
Data root: /home/docker

Rozsah interních síťových adres: defaultní koliduje s lokálními adresami v síti





# DOCKER-COMPOSE



Pro vytvoření všech kontejnerů používáme docker-compose

- nastavení exposovaných portů služeb
- mount volumes do složek
- environmentální proměnné
- build arguments
- závislosti kontejnerů na sobě
- restart policy
- přístup k GPU

Konfigurovatelné přes environ. proměnné

- .envrc
- např LOCAL\_DATA\_DIR=/home/isletnet

```
frontend:
  image:
    ${REPOSITORY}isletnet-frontend:${TAG:-latest}
  build:
    context: .
    dockerfile: frontend/Dockerfile
  args:
    api_url: ${API_URL:?missing-env}
    disable_experimental: "no"
  ports:
    - ${FRONTEND_PORT:?missing-env}:5000
  depends_on:
    - backend
```

# VERZOVÁNÍ



Jednotlivé docker images jsou pojmenované dle schématu

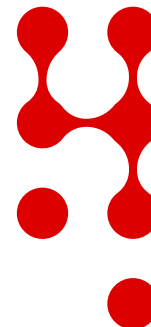
- `docker.iterait.com/isletnet-<komponenta>:<datum>_<verze>`
- Např. `docker.iterait.com/isletnet-frontend:2020-01-29_01`

Více releasů v jednom dni inkrementuje verzi v suffixu

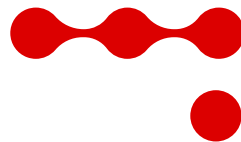
Roll-back na starší verze

Spuštění:

- `REPOSITORY="docker.iterait.com/" TAG="2020-01-29_01" docker-compose up -d`



# NGINX



/etc/nginx/conf.d/isletnet.conf

Dva záznamy s proxy\_pass na nakonfigurované porty kontejnerů

- Frontend: isletnet.com
- Backend: isletnet.com/api

Let's encrypt

- systemd service: certbot-renew.timer



# OVLÁDÁNÍ



Ve složce *isletnet/* (kdekoli v filesystému, jakýkoli uživatel)

- `.envrc`
- `Docker-compose.yml`
- `docker-compose.production.yml`
- záleží na jméně složky, nikoli na jejím umístění (je možné ovládat i jiné instance)

Spuštění

```
REPOSITORY="docker.iterait.com/" TAG="2020-01-29_01" docker-compose up -d [--force-recreate]
```

Vypnutí

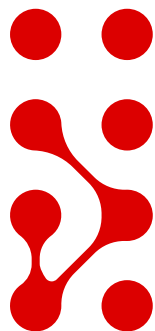
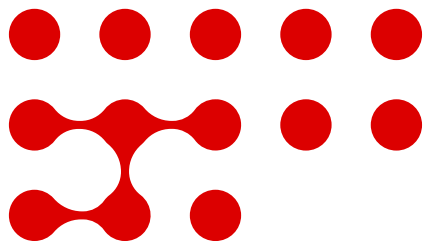
```
REPOSITORY="docker.iterait.com/" TAG="2020-01-29_01" docker-compose down
```

Restart

```
REPOSITORY="docker.iterait.com/" TAG="2020-01-29_01" docker-compose restart
```

Logy

```
REPOSITORY="docker.iterait.com/" TAG="2020-01-29_01" docker-compose logs [-tail=50000] [service name]
```



# 05

## MONITORING

- Monitorování statusu aplikace

# MONITOROVÁNÍ STATUSU APLIKACE



Stav kontejnerů

- REPOSITORY="docker.iterait.com/" TAG="2020-01-29\_01" docker-compose ps
- State: (unhealthy)

Dostupnost frontendu

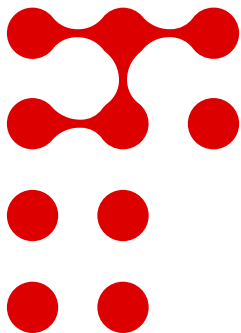
- Ověření HTTP požadavkem na <https://isletnet.com>

End-to-end test (bez frontendu)

- vytvoření triviálního snímku
- odeslání obrázku do REST API backendu <https://isletnet.com/api>
- pokud do několika minut není připravený výsledek -> FAIL

Statistiky přístupů (API)

- [https://isletnet.com/api/statistics/access\\_statistics.xlsx](https://isletnet.com/api/statistics/access_statistics.xlsx)



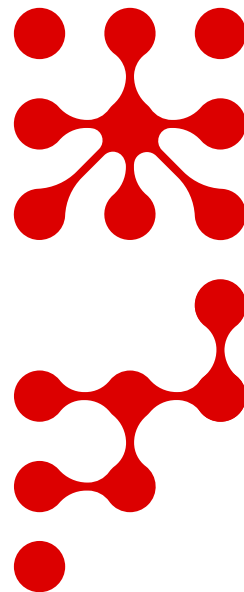
U koho poběží monitoring?  
Jak často ho chceme spouštět?  
- Pozor časté false positive při větší zátěži  
Mailing v případě nedostupnosti?  
Monitoring místa na disku?

## MONITOROVÁNÍ STATUSU APLIKACE

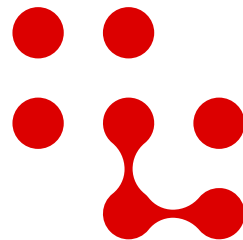
# 06

## TROUBLESHOOTING

Možné scénáře problémů a ■  
jejich řešení



# TROUBLESHOOTING - DOCHÁZÍ MÍSTO NA DISKU



## PROBLÉM

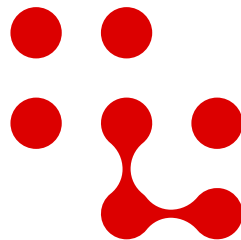
- Nárazově někdo nahrál hodně obrázků
- Fail pruneru
- Příliš mnoho obrázků určených pro trénink
- Příliš mnoho docker logů



## ŘEŠENÍ

- logy (tzn docker partition):  
restart služby (--force-recreate) + promazání kontejnerů
- obrázky:  
restart pruner, příp. manuální odmazání nebo stažení pro trénink

# TROUBLESHOOTING - KONTEJNER SELHAL



## PROBLÉM

- **shepherd** - Problém s komunikací s GPU (/home/isletnet/logs/runner/)
- **shepherd** - Neočekávaná výjimka v postprocessingu (docker logs)
- **backend** - Problém s připojením do sítě (vyřeší se časem sám)



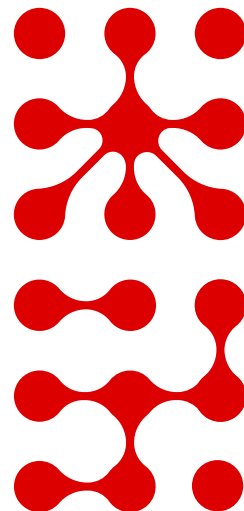
## ŘEŠENÍ

- restart příslušných komponent
- bezstavové - uživatel by restart typicky neměl zaznamenat



## BUDOUCNOST

Jaký bude další ■  
development



# BUDOUCÍ DEVELOPMENT

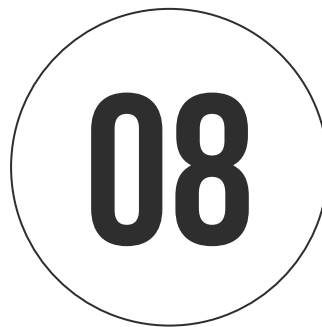
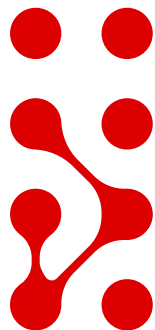
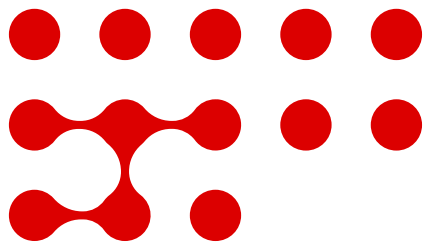


Na prosinec/leden předběžně domluvené rozšíření IsletNetu. Zatím

- rozšíření frontendu
- další neuronové sítě
- přetrénování s více daty
- příprava dat pro vědeckou činnost (3D ostrůvky, spinacle, ...)

Součástí může být

- update použitých technologií
- integrace na monitorovací systémy IKEM
- API/dashboard se statusem jednotlivých komponent
- ...
- Co by se Vám hodilo?



## SHRNUTÍ

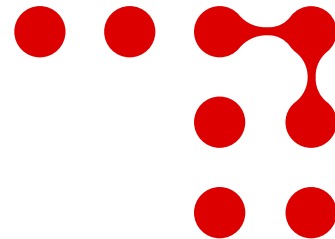
- Závěr a prostor pro vaše otázky

# SHRNUTÍ



- Architektura s microservices
- REST API -> napojení vlastních služeb
- Většinu problémů vyřeší docker-compose restart
- Na přelomu roku možné technické změny (např. pro potřeby správy/monitoringu)
- Výzkumná část může snadno běžet přímo v IKEMu
- Možný běh v cloudu

## DISKUZE



# DÍKY ZA POZORNOST

Petr Bělohlávek

+420 731 012 334

[petr.belohlavek@iterait.com](mailto:petr.belohlavek@iterait.com)

