

# EXPERIMENTÁLNA WEBOVÁ APLIKÁCIA PRE BIOMETRICKÝ AUTENTIFIKAČNÝ SERVER



**Vedúci práce:** Ing. Pavol Marák PhD.



**Autor práce:** Ján Vladár

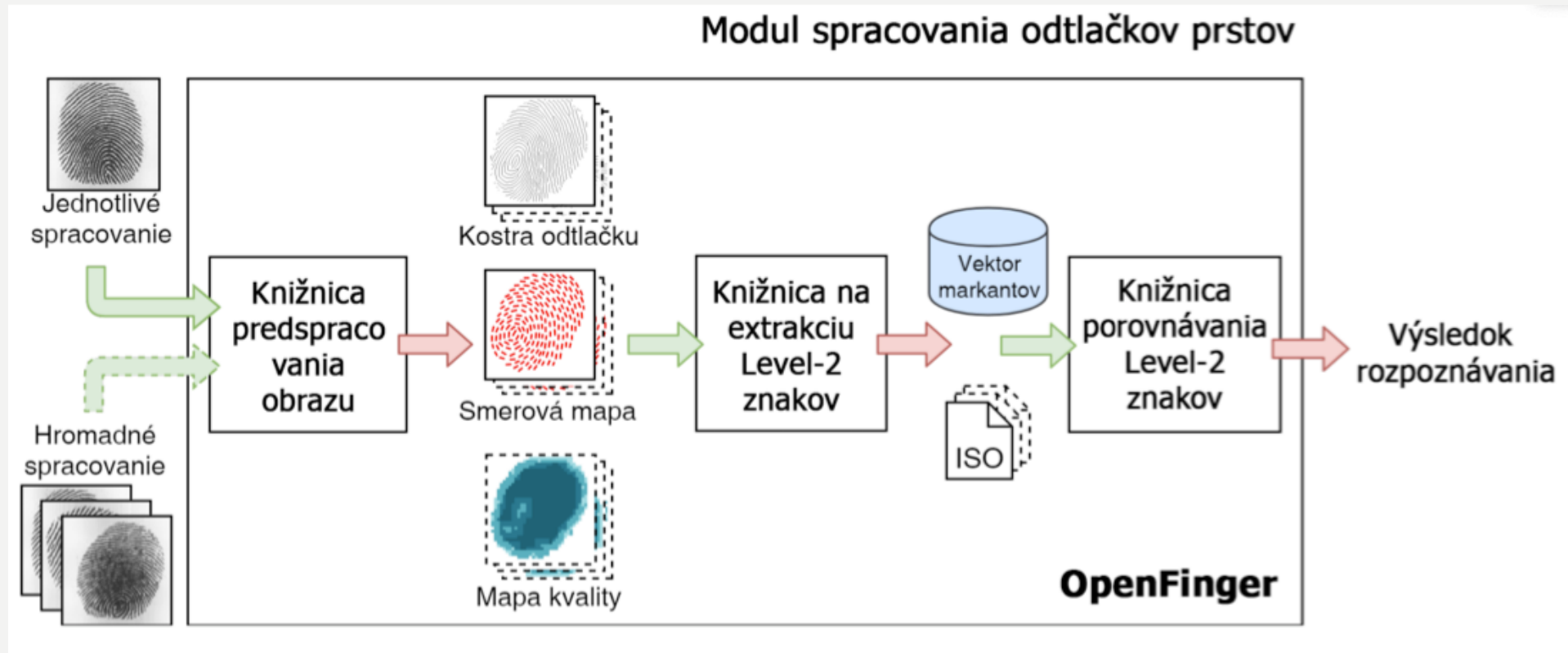
# OBSAH

1. Riešený problém a motivácia
2. Návrh riešenia
3. Dosiahnuté výsledky
4. Sumarizácia

# MOTIVÁCIA

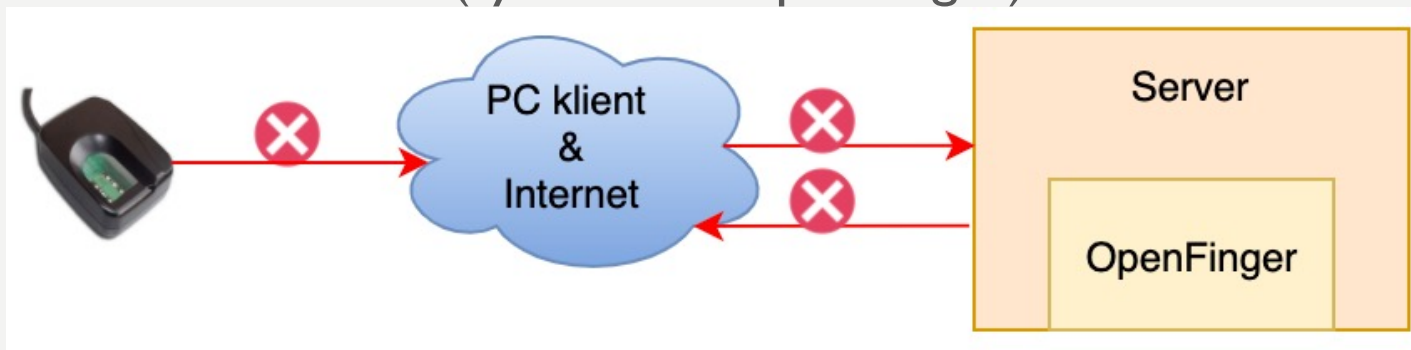
- Existujúci multimodálny systém OpenFinger dostupný cez PC sieť
- OpenFinger obsahuje okrem iného aj modul spracovania odtlačkov prstov, ktorý vie vykonávať rôzne operácie: .
  - predspracovanie
  - extrakcia Level-2 znakov
  - verifikácia
  - identifikácia
- **Systém má chýbajúce používateľské aj experimentálne rozhranie**

# MODUL SPRACOVANIA ODTLAČKOV PRSTOV



# RIEŠENÝ PROBLÉM

- Rozhranie pomocou webovej aplikácie, ktorá umožní :
  - Šifrovanú komunikáciu so serverom cez SSL/TCP soket, server má vygenerovaný samopodpísaný SSL/TLS certifikát
  - Registráciu osoby do databázy užívateľov pomocou odtlačku prstu
  - Identifikáciu a verifikáciu užívateľa podľa odtlačku prstu
  - Nastavovanie parametrov predspracovania obrazu systému OpenFinger s následným zobrazením predspracovaného obrazu v rôznych fázach
  - Zobrazenie zoskenovaného alebo nahraného odtlačku
  - Komunikácia so snímačom odtlačkov prstov
  - Komunikácia so serverom (systémom OpenFinger)



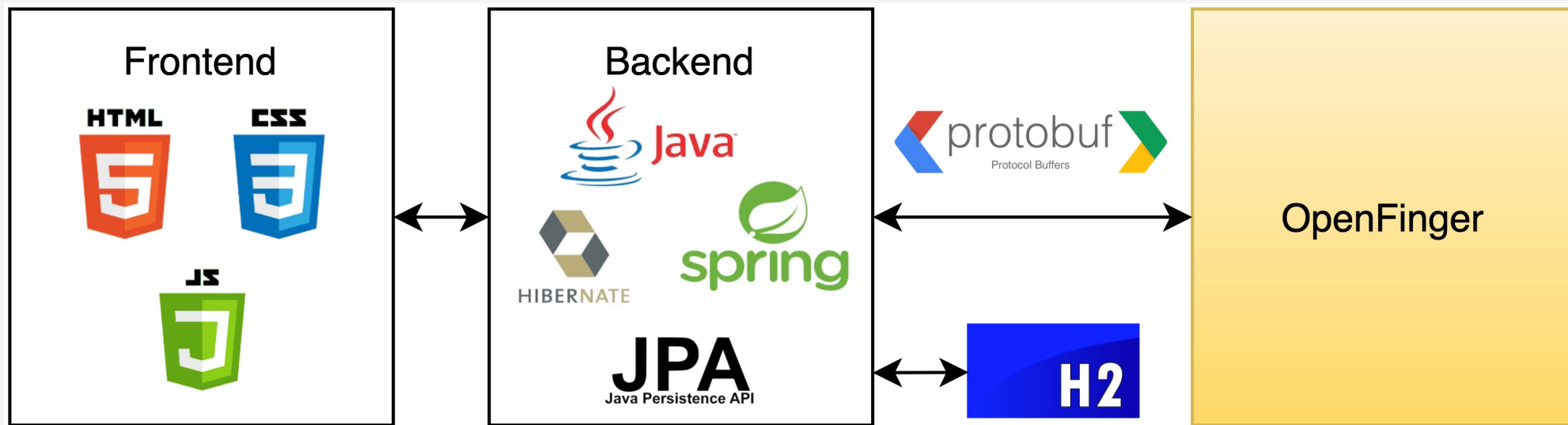
# NÁVRH A POŽIADAVKY

- Prepojenie skenera s web. aplikáciou
- Možnosť nahrat' odtlačok z disku
- Serializácia a deserializácia požiadavky, aby server vedel určiť o ktorú požiadavku sa jedná
  - Registrácia odtlačku do databázy
  - Pridanie ďalšieho odtlačku používateľovi do databázy
  - Verifikácia užívateľa s jeho už uloženými odtlačkami
  - Identifikácia používateľa so všetkými uloženými odtlačkami
  - Predspracovanie odtlačku aj s nastavením špecifických parametrov
  - Zobrazenie zoskenovaného odtlačku

# NÁVRH A POŽIADAVKY

- Individuálne podľa typu požiadavky vyhodnocovať a zobrazovať rôzne druhy požiadaviek a pracovať s databázou
- Minimalizácia spotreby dát a čo najrýchlejšia komunikácia so serverom
- Minimálna veľkosť posielaných dát na spracovanie systémom OpenFinger

# POUŽITÉ TECHNOLOGIE



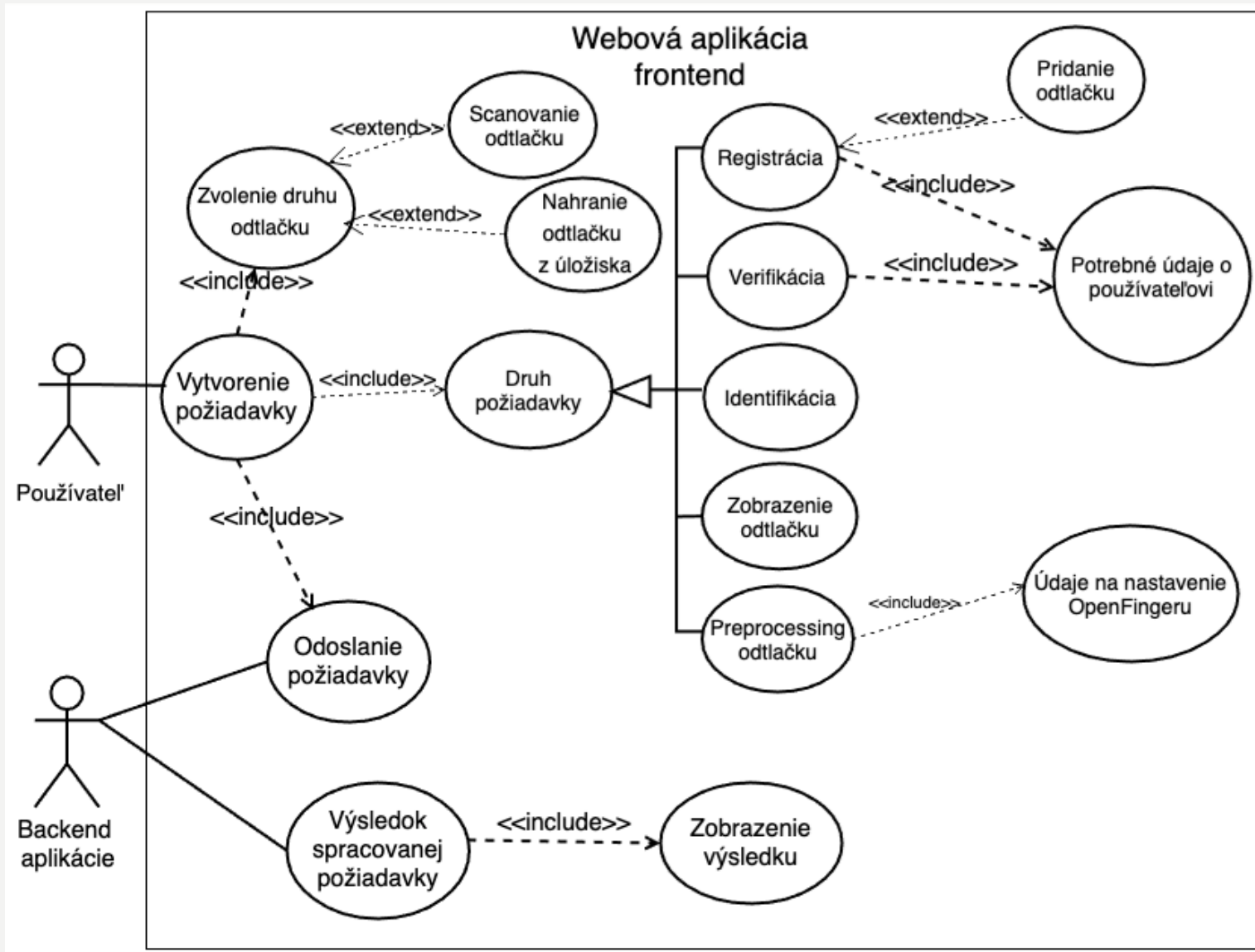
**HTML, CSS, Javascript** – tvorba používateľského rozhrania

**Java, framework Spring** – zabezpečujú všetko čo beží na pozadí aplikácie, komunikujú s databázou, posiela aj prijímajú dáta na a z frontendu

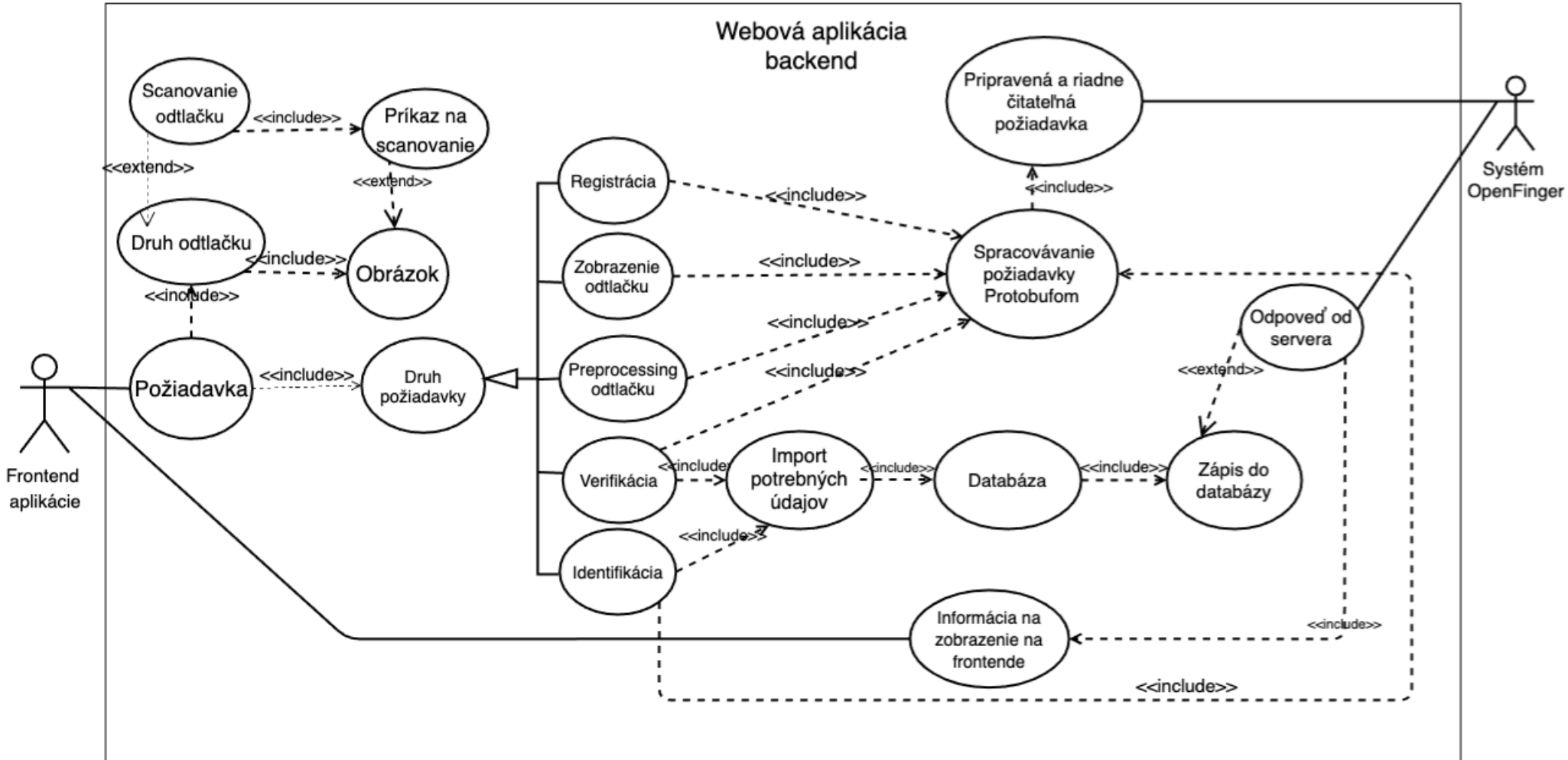
**JPA, Hibernate, H2 databáza** – JPA definuje špecifikáciu, ktorá nám umožňuje mapovanie Java objektov na tabuľky v databáze, Hibernate je špecifická implementácia JPA.



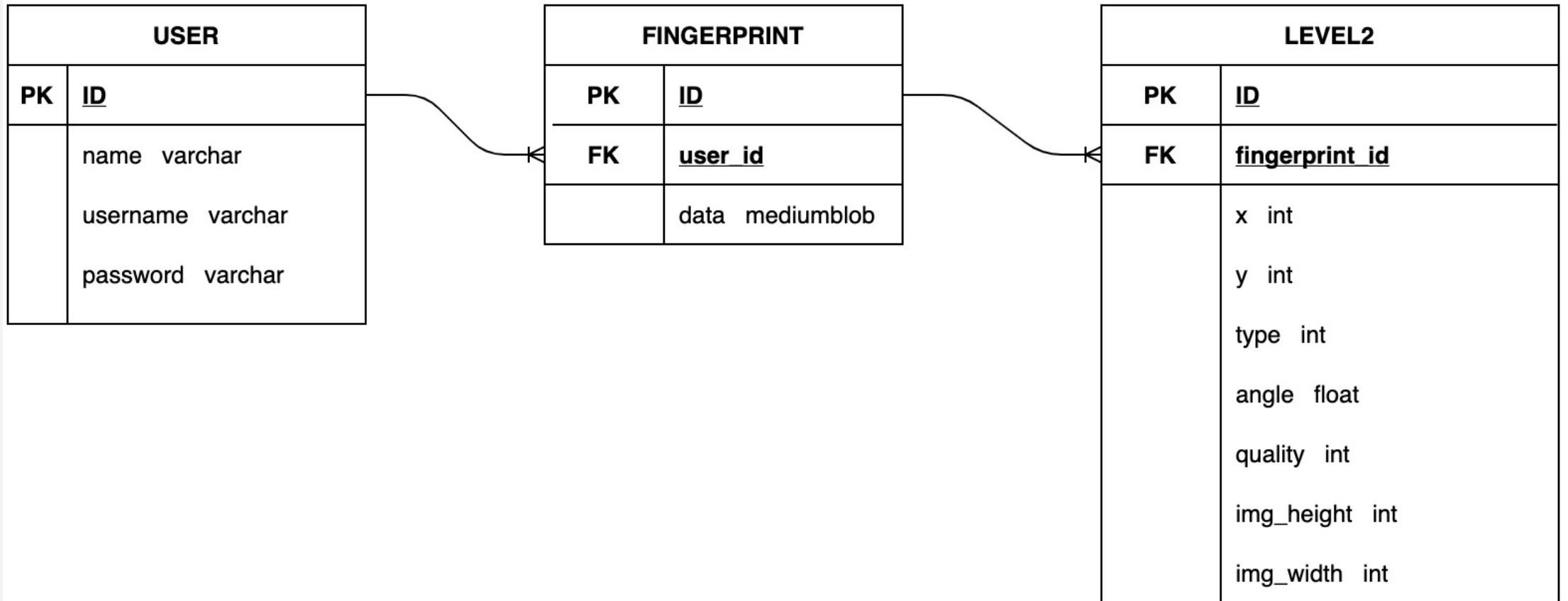
# FRONTEND



# BACKEND



# DATABÁZA



# DOSIAHNUTÉ VÝSLEDKY

STU  
SLOVENSKÁ TECHNICKÁ  
UNIVERZITA V BRATISLAVE

Scanovanie odtlačkov

Požiadavka Vyber požiadavku

SKENOVAŤ

Nahrание odtlačkov

Požiadavka Predspracovanie odtlačku

Gabor Lambda 9

Gabor Sigma 3

Block size 13

Nahrať odtlačok Vybrať súbory Nie je vybratý žiadny súbor

ODOŠLI

Požiadavka Kostra odtlačku

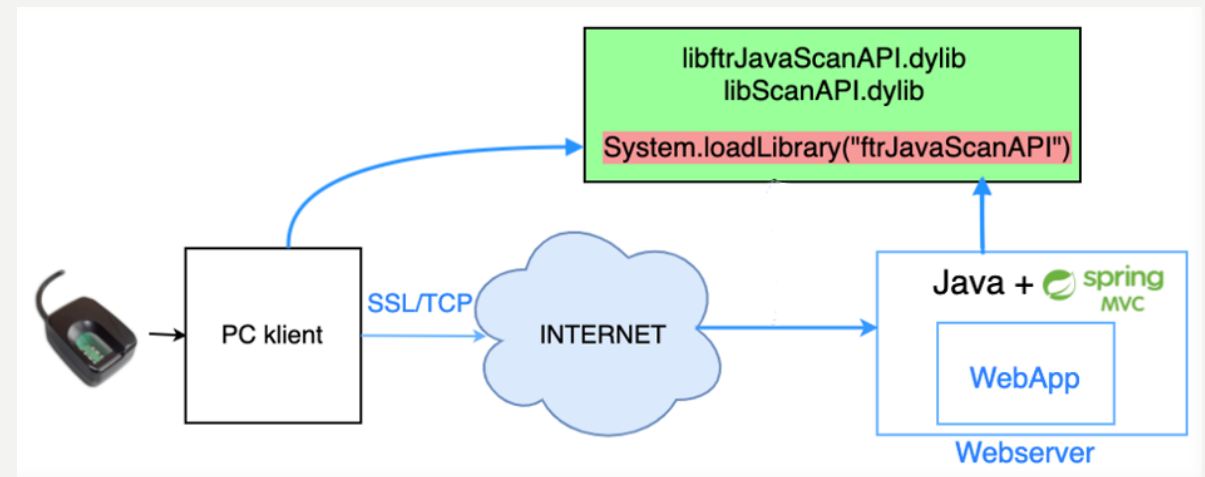
Výsledok požiadavky



Výsledok požiadavky predpracovania a zobrazenie kostry odtlačku prstu

# SPRACOVANIE OBRAZU

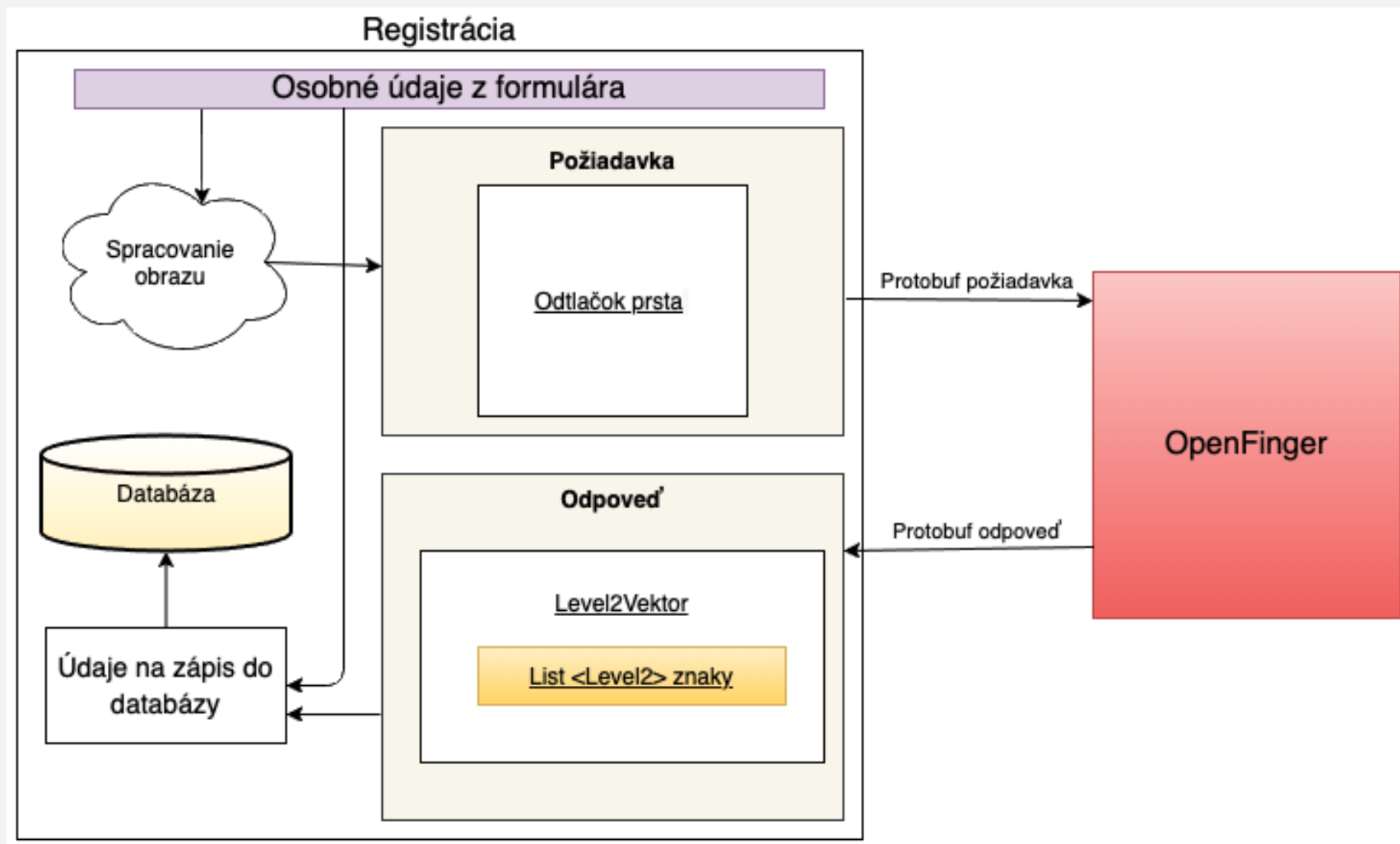
- Komunikácia so senzorom Futronic FS-80H
  - pomocou JNI (Java Native Interface)
- Spracovanie a získanie dát zo senzora
  - spracovanie obrázku z formy bajtov až po serializáciu do premennej typu ByteString
- Získanie dát z nahraného obrázku odtlačku
  - spracovanie obrázku rôznych typov (png, jpg, bmp, ...) až po serializáciu do premennej typu ByteString



# TYPY POŽIADAVIEK

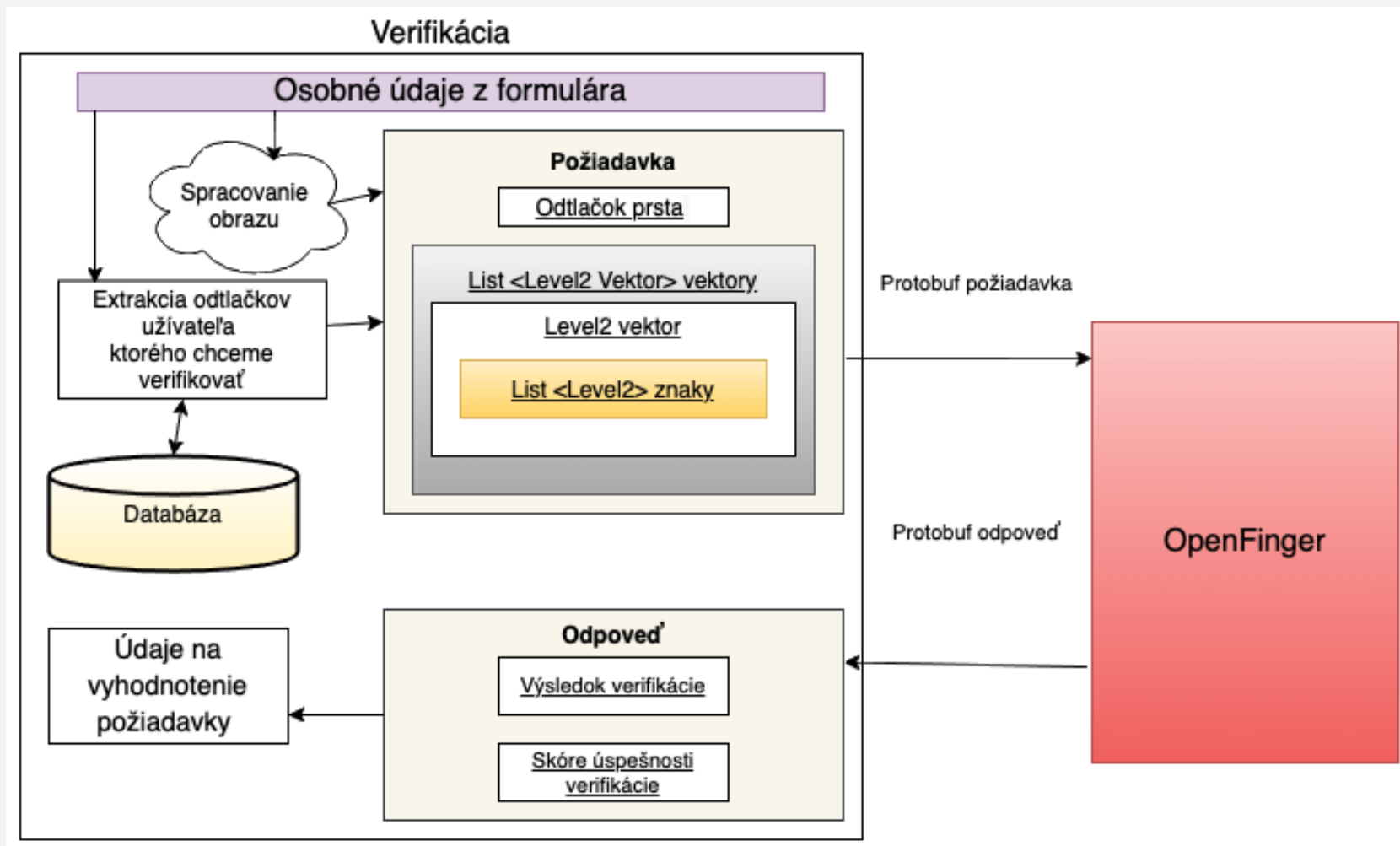
- Registrácia
- Verifikácia
- Identifikácia
- Predspracovanie odtlačku
- Zobrazenie odtlačku

# REGISTRÁCIA




# VERIFIKÁCIA

Porovnávam: Aktuálny odtlačok **VS** Všetky odtlačky daného používateľa





# REGISTRÁCIA A VERIFIKÁCIA



SLOVENSKÁ TECHNICKÁ  
UNIVERZITA V BRATISLAVE

## Scanovanie odtlačkov

Požiadavka

Meno a priezvisko

Prihlasovacie meno

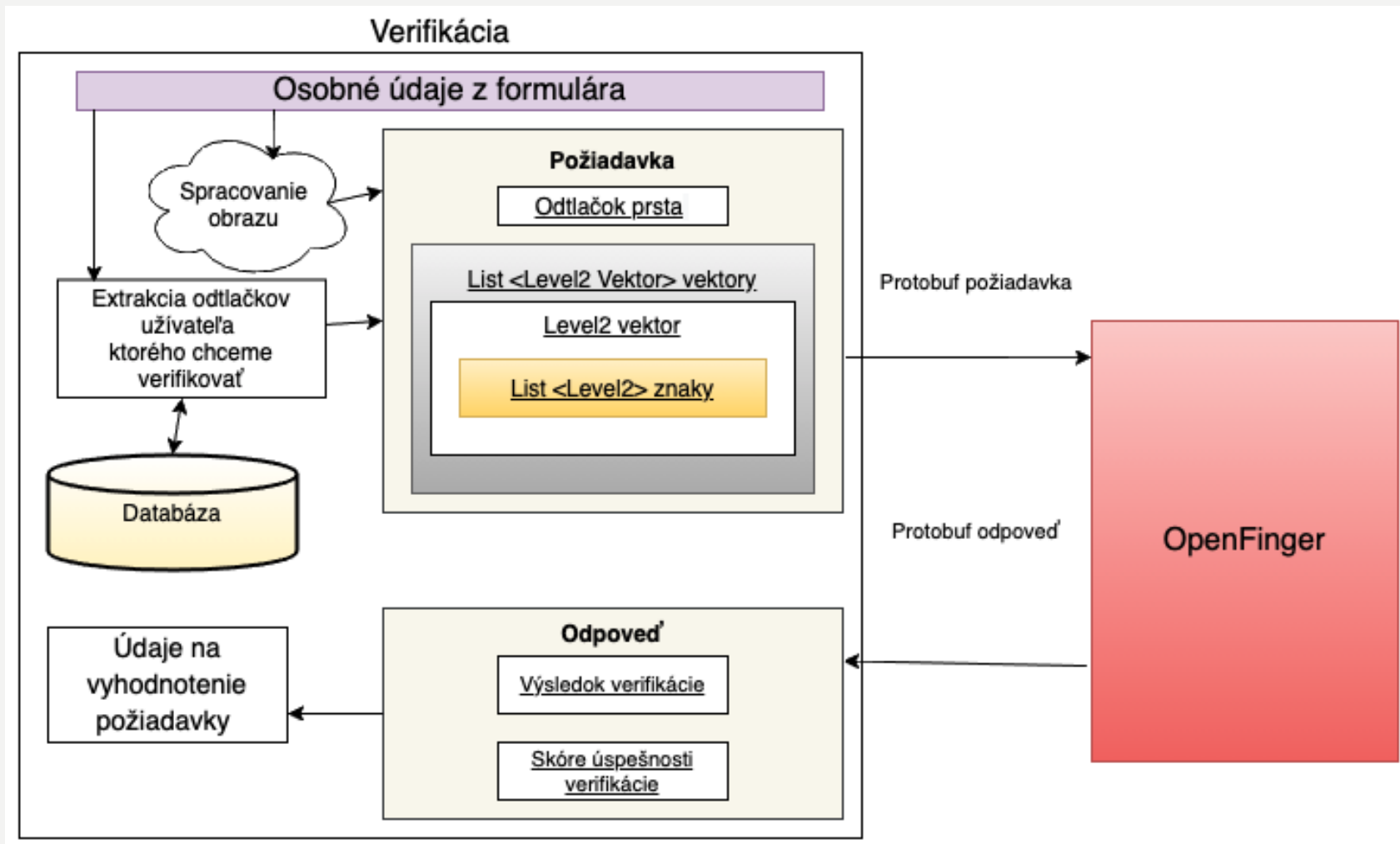
**SKENOVAŤ**

Nahrание odtlačkov ☐

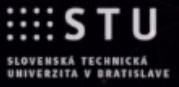
## Výsledok požiadavky

# IDENTIFIKÁCIA

Porovnávam: Aktuálny odtlačok **VS** Všetky odtlačky z databázy !



# IDENTIFIKÁCIA



SLOVENSKÁ TECHNICKÁ  
UNIVERZITA V BRATISLAVE

## Scanovanie odtlačkov

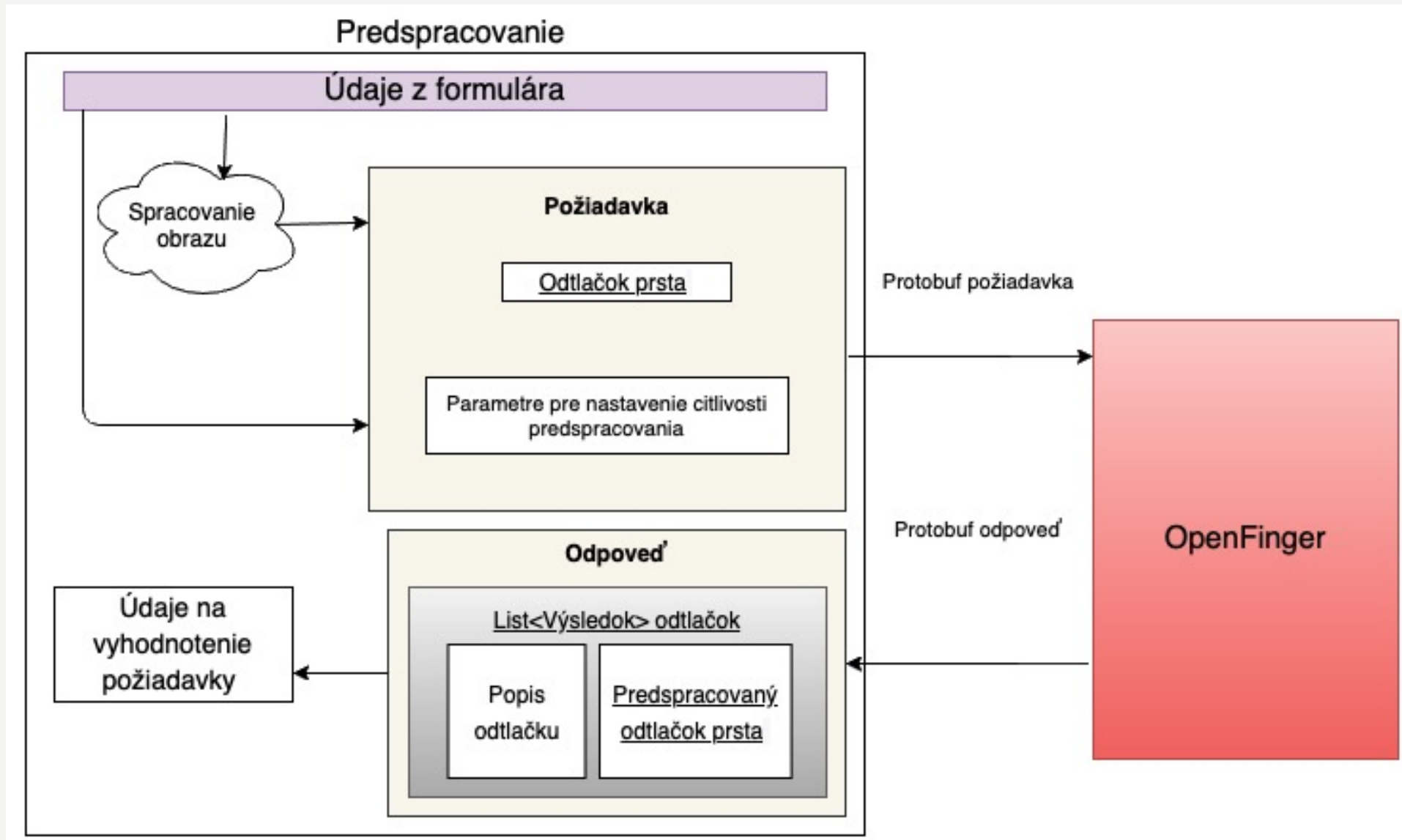
Požiadavka

**SKENOVAŤ**


Nahrание odtlačkov ☐

## Výsledok požiadavky

# PREDSPRACOVANIE ODTLAČKU



# PREDSPRACOVANIE ODTLAČKU



SLOVENSKÁ TECHNICKÁ  
UNIVERZITA V BRATISLAVE

## Scanovanie odtlačkov

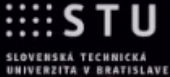
Požiadavka

**SKENOVAŤ**

Nahrание odtlačkov ☐

## Výsledok požiadavky

# ZOBRAZENIE ODTLAČKU



SLOVENSKÁ TECHNICKÁ  
UNIVERZITA V BRATISLAVE

## Scanovanie odtlačkov

Požiadavka

Zobraz odtlačok

▼

SKENOVAŤ

Nahrание odtlačkov

☐

Výsledok požiadavky

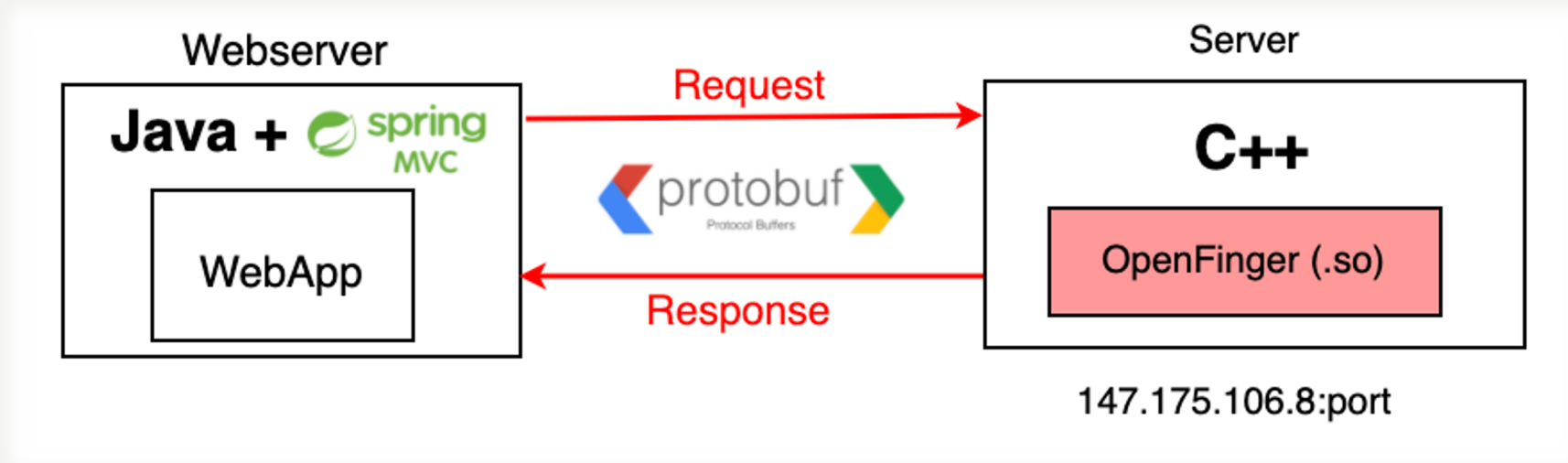
42



# SERIALIZÁCIA A DESERIALIZÁCIA

- Soketová komunikácia webovej aplikácie a biometrického servera
  - Komunikácia so serverom pomocou *Google Protocol Buffers ( .proto )*

Prečo Protobuf ? Pretože chceme komunikovať medzi rôznymi programovacími jazykmi. Pomocou Protobufu vieme medzi Javou a C++ bezproblémovo posielat' a následne prečítať zaseriazované triedy cez vopred zadefinovaný *.proto* súbor.



# UKÁŽKA SERIALIZÁCIE A DESERIALIZÁCIE

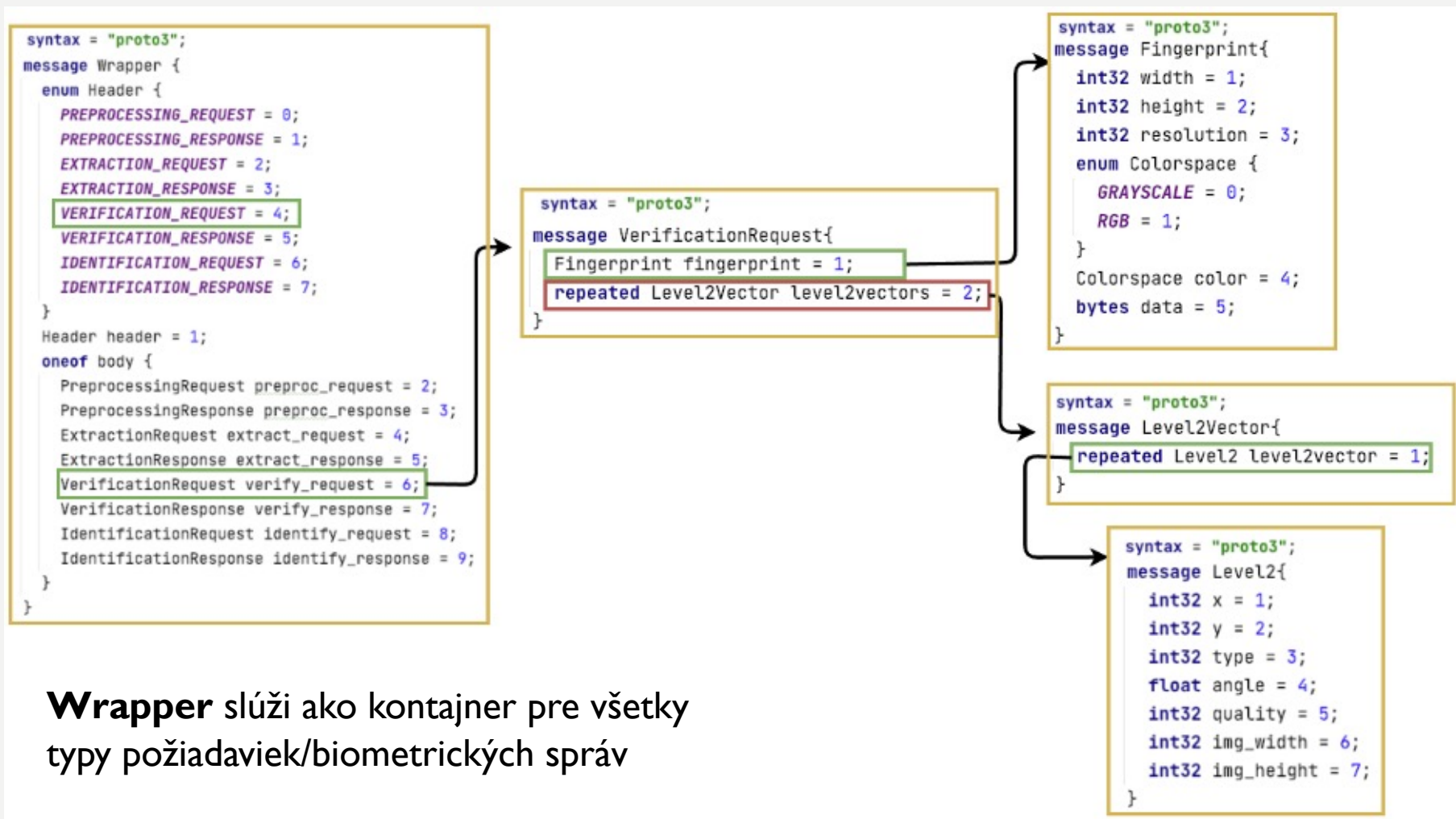


Diagram Protobuf triedy Wrapper aj s jej ďalšími zapúzdrenými údajmi pri verifikačnej požiadavke



# SUMARIZÁCIA

- Implementácia komunikácie senzora Futronic FS-80H s aplikáciou
- Možnosť nahrania odtlačku prsta z disku zariadenia
- Úspešne spracovanie, serializovanie a následne zobrazenie rôznych typov obrazov na strane klienta
  - Registrácia a následne pridanie odtlačkov ku danému užívateľovi
  - Verifikácia a identifikácia užívateľa na základe porovnávania jeho odtlačku s ďalšími
  - Predspracovanie obrazu kedy si vie používateľ pozerat' obrazy v 6 rôznych fázach predspracovania
  - Zobrazenie odtlačku prstu na strane klienta
- Aplikácia vie úspešne vyhodnocovať, ukladať do databázy a zobrazovať odpovede na požiadavky na strane klienta, ktoré sú vždy iného typu
- Komunikácia s minimálnou spotrebou dát, ktorú sme zabezpečili posielaním zaserializovaných tried typu Google Protocol Buffers
- Minimalizácia veľkosti posielaných dát vďaka registračnému procesu, ktorý nám vracia len Level-2 znaky

# OTÁZKY OPONENTA

- Aký je rozdiel medzi identifikáciou a verifikáciou osoby?
- Prečo ste zvolili jazyk Java a framework Spring pre vašu aplikáciu? Prečo ste nepoužili napríklad jazyk Python alebo C++?