

Balkanski špijun

Autori:

Tomislav Petričević

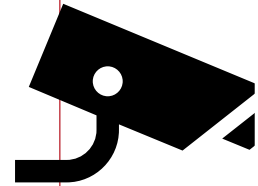
Ivan Andrić

11. listopad 2021

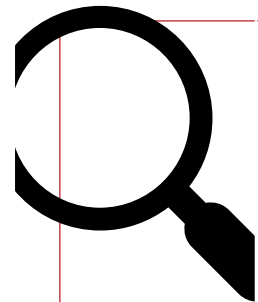


O čemu je riječ?

- Zahtjevi
- Izazovi
- Tehničko rješenje
- Arhitektura
- Rezultati i statistike



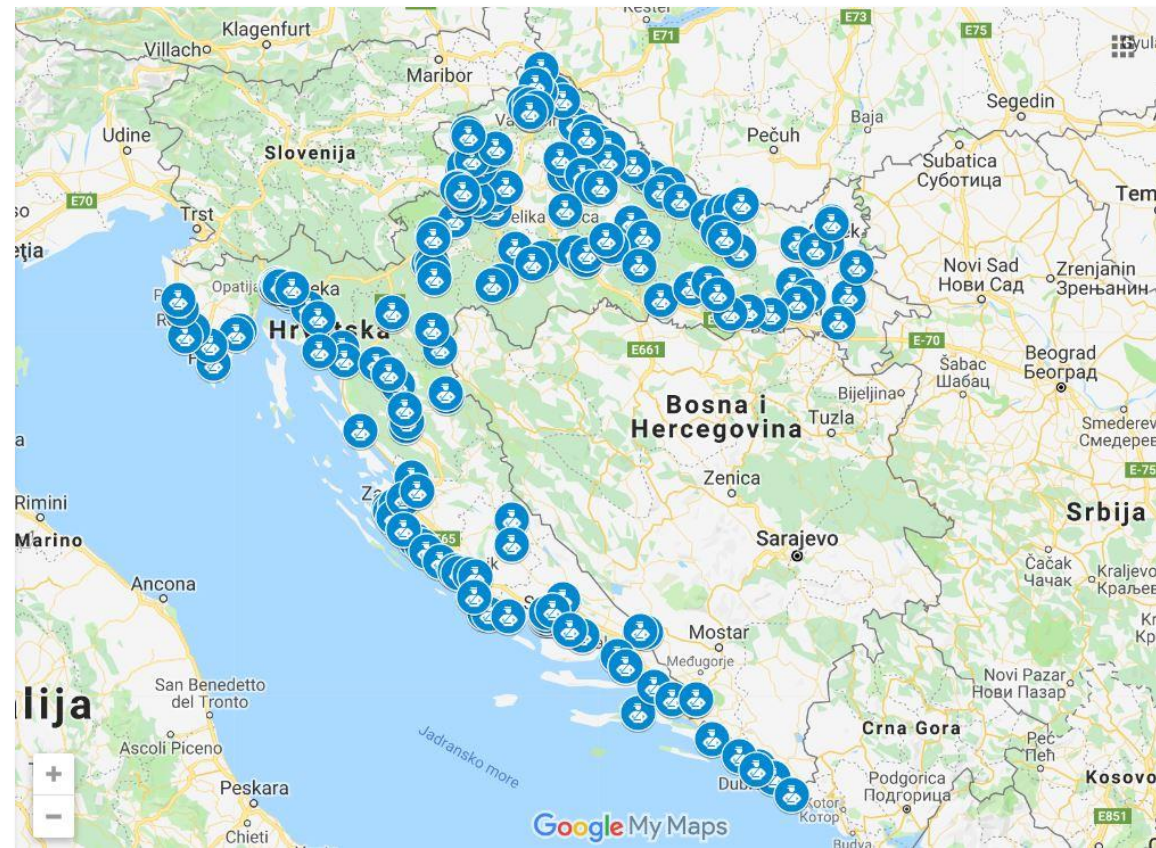
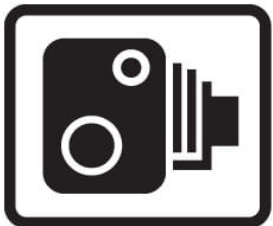
Obrada podataka s
kamera



Provjera
registarskih pločica

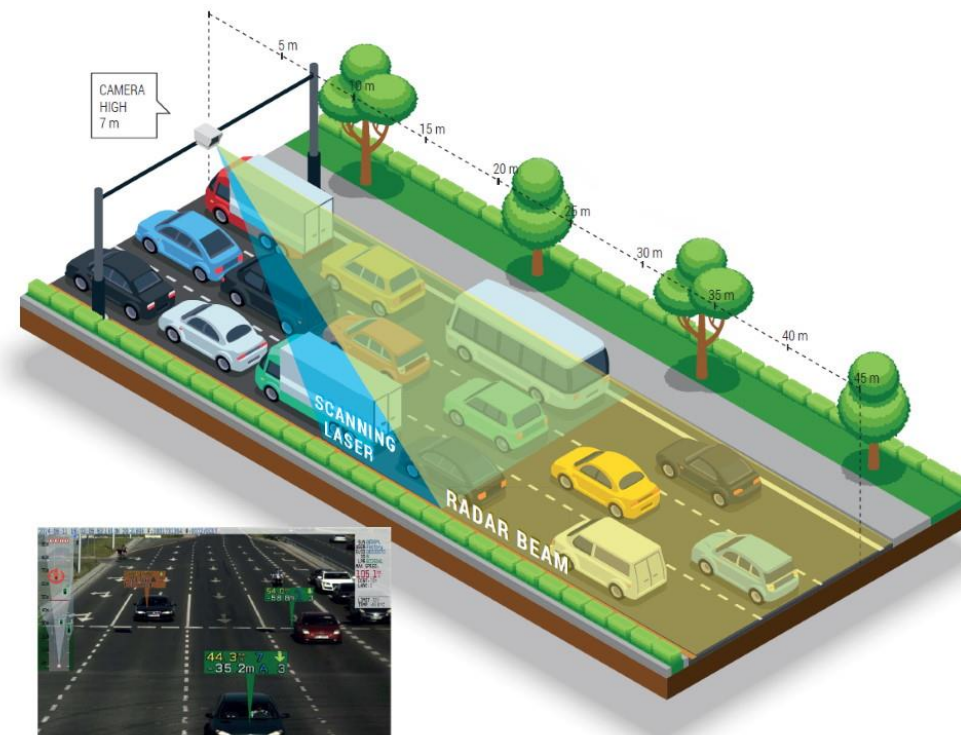
Kamere na hrvatski prometnicama

- >300 fiksnih kamera
 - Plan je x10
- >100 tronožaca
- >10000 vozila dnevno po kameri
- >1400 prekršaja dnevno
- Prekoračenje brzine je svaki drugi prometni prekršaj



Kamere u brojevima

- radarski snop pokriva do 6 prometnih traka, radi 24/7
- vozilo se snima na 100m udaljenosti
- Snima 32 vozila u oba smjera odjednom
- mogu izmjeriti brzine do 300 km/h
- 20MP senzor



Kamera u slici

ROAD PLAN

CAMERA VIEW IMAGE

VIOLATION VEHICLE IMAGE

DRIVER IMAGE

NUMBER PLATE IMAGE

EVIDENCE ID: 0000106486	LOCATION: A1. Jerozolimskie	NUMBER PLATE: WW168 [PL]
DATE/TIME: 2017-01-27 16:15:27	GPS POSITION: 52.1869E 20.9077N	VEHICLE ID: 32
DEVICE MODEL: SmartEye	VIOLATION TYPE: SPEED	VEHICLE MAKE: SKODA
DEVICE TYPE: ST-1	SPEED LIMIT: 70 km/h	VEHICLE MODEL: OCTAVIA
SERIAL NO: 0121PL	LIGHT STATUS: RED [754 ms]	VEHICLE COLOR: PLAVA
FIRMWARE VER: 3.7	MEASURED SPEED: 76.3 km/h	VEHICLE TYPE: CAR
FIRMWARE CRC: 43FCA328	FRAME NO: 1 OF 13	
FILENAME: 170127_161527_0121PL_0000106486.psev	FRAME ID: 023432356	
	CRC: 42bdefaf1d620b142963889c188e61aef17bffc6896d1874	



Obrada podataka s kamera

Radi se o detektiranju vozila u prekoračenju brzine te obrade istih u svrhu daljnjeg procesuiranja prekršaja.

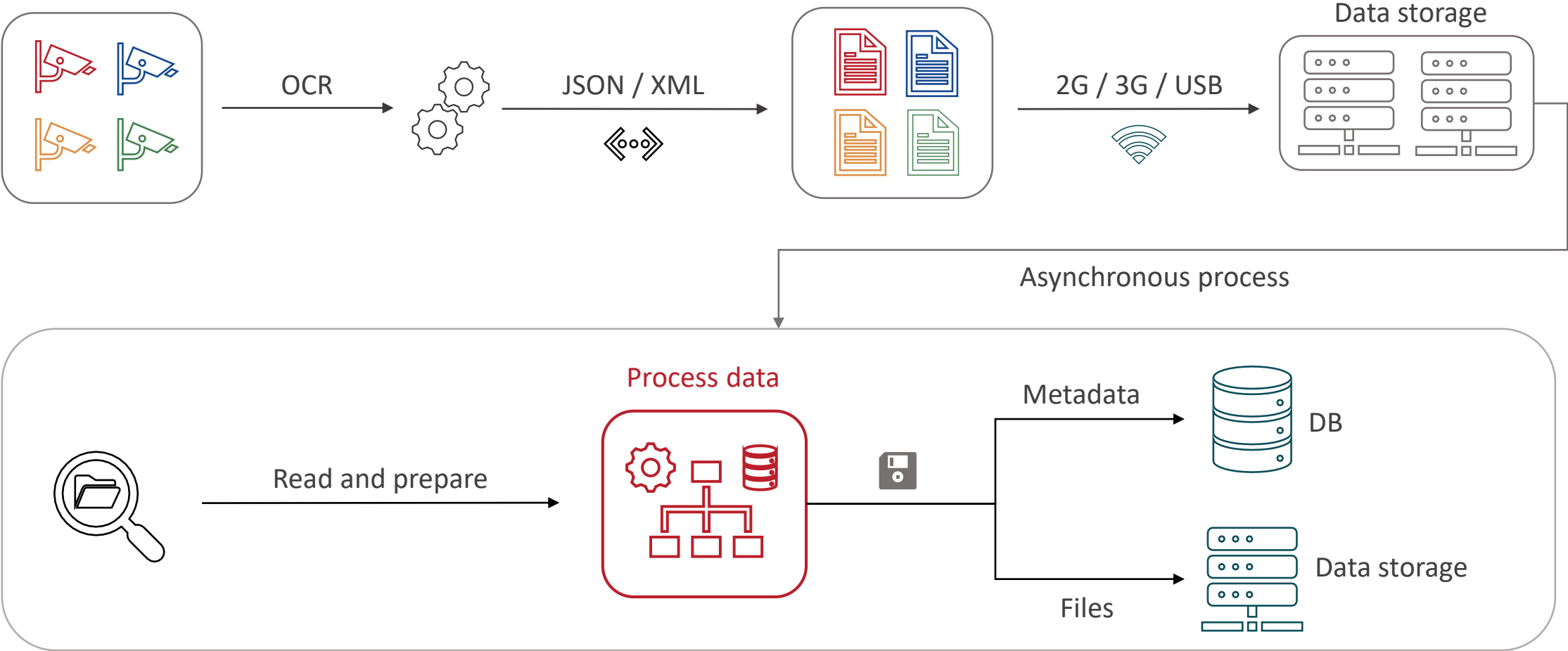
Korisnički zahtjevi

- Brzo
- Skalabilno
- Konfigurabilno
- Modularno
- Jednostavno za integraciju
- Jednostavno za održavanje

Izazovi

- Količina podataka
- Skladištenje podataka
- Odabir tehnologija i procesa
- Mapiranje podataka
- Monitoriranje
- Održavanje

Proces dohvata i obrade podataka



Spring Integration

Što je to?

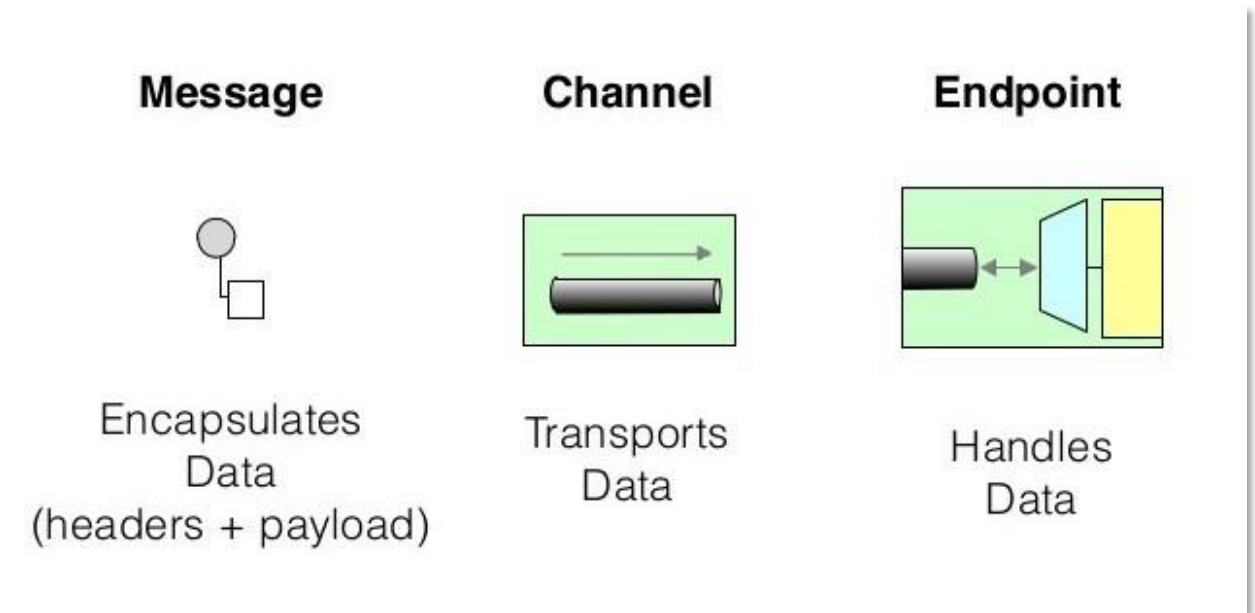
Spring rješenje za razmjenu poruka

Zašto?

- Lagan
- Jednostavan za integraciju
- Mnoštvo out-of-the-box komponenti

Osnovne komponente

- Poruka
- Kanal poruke
- Krajnja točka



Implementacija

Koraci:

- 1) Kreirati tablice na bazi
- 2) Kreirati kanale poruka
 - MetadataStore
 - ChannelMessageStore
- 3) Definirati IntegrationFlow
 - Izvor poruke (supplier)
 - Krajnja točka (endpoint)
- 4) Kreiranje taska za otkrivanje novih snimki
- 5) Kreiranje i slanje poruke
- 6) Obrada poruke

```
@Bean
IntegrationFlow mainIntegrationFlow() {
    return IntegrationFlows.from(supplier, consumer)
        .handle(this::processMessage)
        .get();
}

@Override
void readMessageSource() {
    // discover files using FileVisitor
    // create and add message to queue
}

@Override
void processMessage(Message<?> message) {
    // validate and process message
}
```

I FileVisitor<T>		
m	postVisitDirectory(T, IOException)	FileVisitResult
m	preVisitDirectory(T, BasicFileAttributes)	FileVisitResult
m	visitFile(T, BasicFileAttributes)	FileVisitResult
m	visitFileFailed(T, IOException)	FileVisitResult

C SimpleFileVisitor<T>		
m	postVisitDirectory(T, IOException)	FileVisitResult
m	preVisitDirectory(T, BasicFileAttributes)	FileVisitResult
m	visitFile(T, BasicFileAttributes)	FileVisitResult
m	visitFileFailed(T, IOException)	FileVisitResult

C AbstractFileVisitor		
f	log	Logger
f	filePatternFilter	Predicate<File>
f	lastModifiedFilter	Predicate<File>
f	terminate	BooleanSupplier
f	files	List<Path>
m	addFile(Path)	void
m	getFiles()	List<Path>
m	visitFile(Path, BasicFileAttributes)	FileVisitResult
m	visitFileFailed(Path, IOException)	FileVisitResult
m	visitRecording(File)	void

C CordonFileVisitor		
f	discoverPath	Path
m	visitRecording(File)	void

C PolcamFileVisitor		
f	discoverPath	Path
m	visitRecording(File)	void

C GatsoFileVisitor		
f	discoverPath	Path
m	getRecordingType(File)	GatsoRecordingType
m	isViolation()	Predicate<File>
m	visitRecording(File)	void

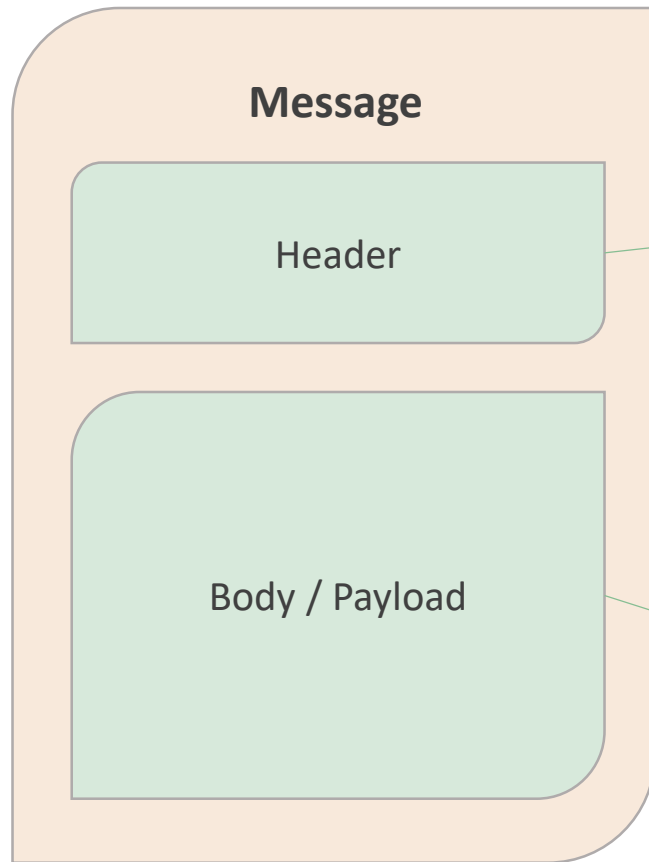
C AbstractMessageHandler		
f	msgChannel	PollableChannel
f	dlqChannel	PollableChannel
f	plqChannel	PollableChannel
f	msgReceiver	DynamicPeriodicMessageReceiver
f	dlqReceiver	DynamicPeriodicMessageReceiver
f	plqReceiver	DynamicPeriodicMessageReceiver
f	msgPollPeriod	Duration
f	dlqPollPeriod	Duration
f	plqPollPeriod	Duration
f	msgTTL	Duration
f	dlqTTL	Duration
f	plqTTL	Duration
f	dlqExpiration	Integer
f	plqExpiration	Integer
f	channelMessageStore	ChannelMessageStore
f	concurrentMetadataStore	ConcurrentMetadataStore
m	addMessageToQueue(String, String)	void
m	genericDeadLetterQueueFlow()	IntegrationFlow
m	genericParkingLotQueueFlow()	IntegrationFlow
m	init()	void
m	moveToDeadLetterQueue(Message<?>, String, String)	void
m	postRemoveMessage(Message<?>)	void
m	processDeadLetterQueue(Message<?>)	void
m	processMessage(Message<?>)	void
m	processMessageQueue(Message<?>)	void
m	processParkingLotQueue(Message<?>)	void
m	readMessageSource()	void
m	removeMessage(Message<?>)	void
m	scheduleMessageRead()	void

C GatsoRecordingHandler		
f	discoverPath	Path
f	integrationFacade	GatsoIntegrationFacade
m	gatsoRecordingDlqFlow()	IntegrationFlow
m	gatsoRecordingFlow()	IntegrationFlow
m	gatsoRecordingPlqFlow()	IntegrationFlow
m	processMessage(Message<?>)	void
m	readMessageSource()	void

C PolcamRecordingHandler		
f	discoverPath	Path
f	integrationFacade	PolcamIntegrationFacade
m	polcamRecordingDlqFlow()	IntegrationFlow
m	polcamRecordingFlow()	IntegrationFlow
m	polcamRecordingPlqFlow()	IntegrationFlow
m	processMessage(Message<?>)	void
m	readMessageSource()	void

C CordonRecordingHandler		
f	discoverPath	Path
f	integrationFacade	CordonIntegrationFacade
m	cordonRecordingDlqFlow()	IntegrationFlow
m	cordonRecordingFlow()	IntegrationFlow
m	cordonRecordingPlqFlow()	IntegrationFlow
m	processMessage(Message<?>)	void
m	readMessageSource()	void

Struktura poruke



DeathHeader		
f	queue	String
f	reason	String
f	creationTime	Long
f	time	Long
f	count	Integer
f	totalCount	Integer

RecordingMessage		
f	speedometer	String
f	path	String
m	from(SpeedometerType, Path)	RecordingMessage
m	getPath()	String
m	getSpeedometer()	String

Struktura tokova

FIFO princip

3 toka

Parametri



Main Queue

- Procesuira poruku na daljnju obradu

Dead Letter Queue

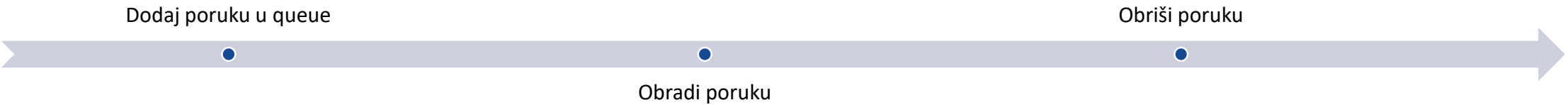
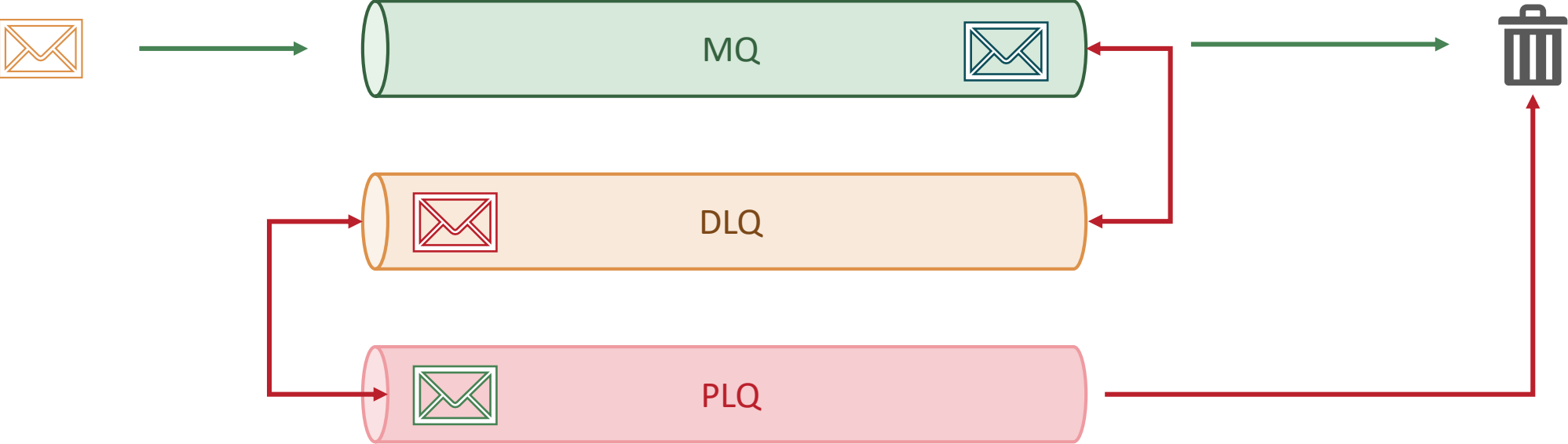
- Zaprima neuspjele obrade iz glavnog toka

Parking Lot Queue

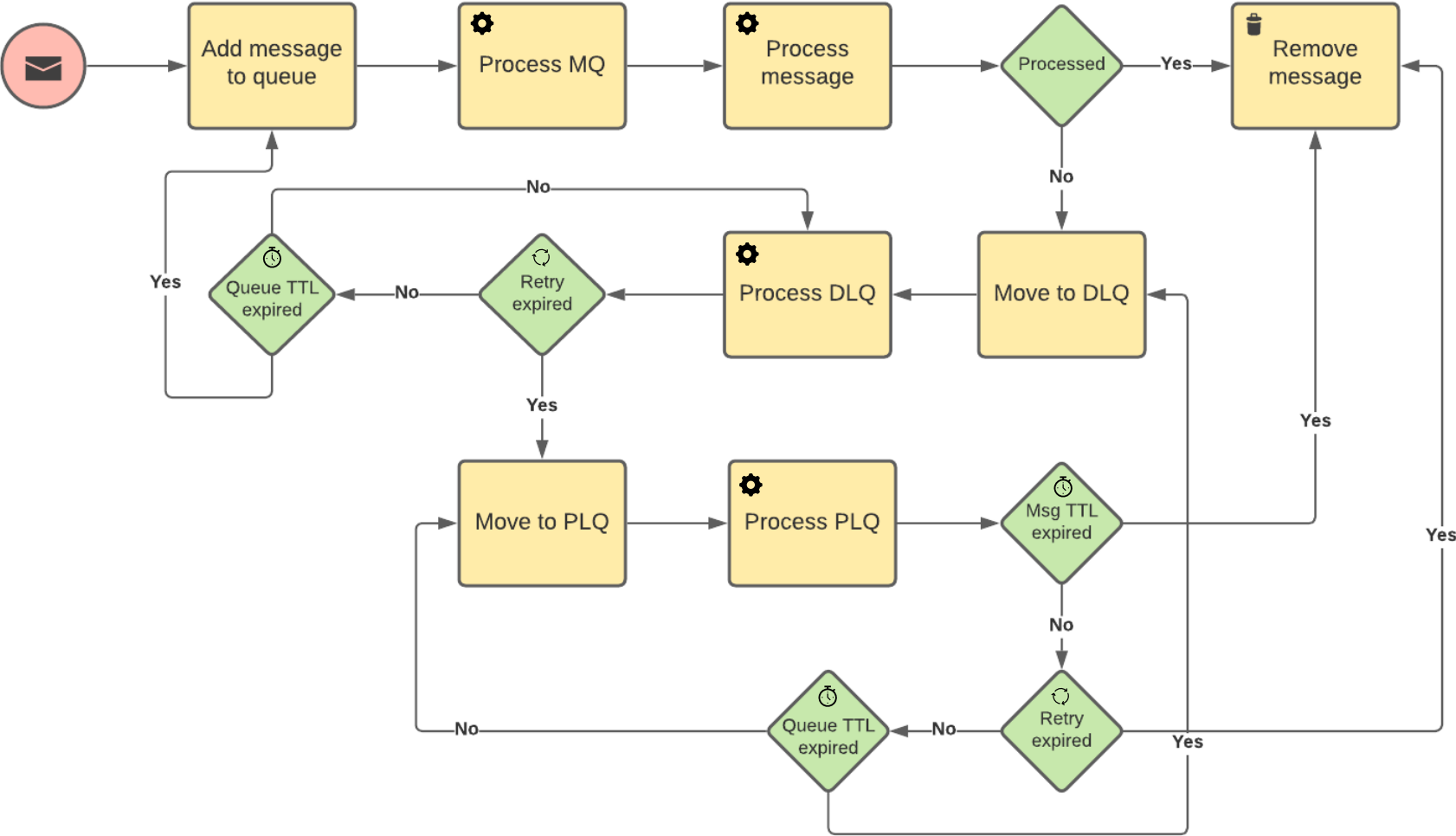
- Zadržava poruke koje se više puta nisu uspjele obraditi

Pool period / TTL / broj pokušaja

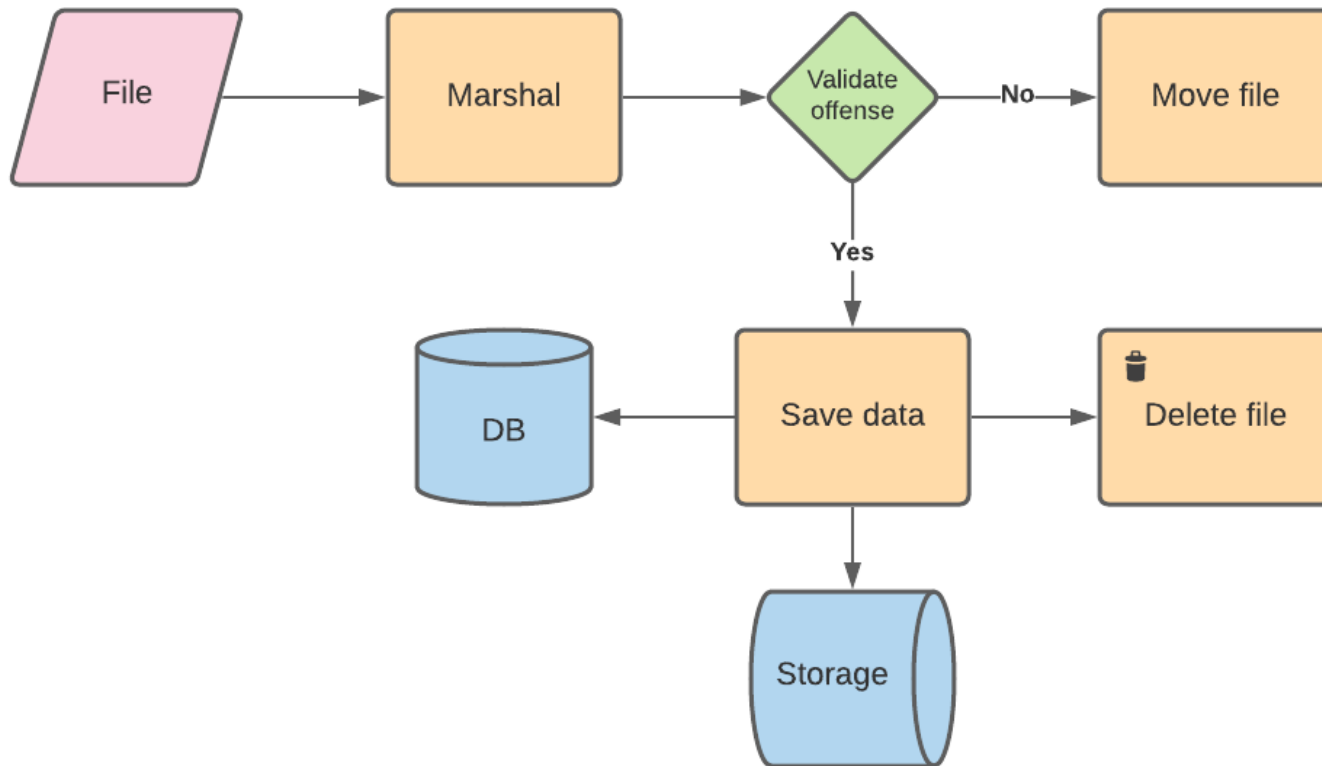
Proces poruke



Dijagram toka



Obrada podataka



Metapodaci

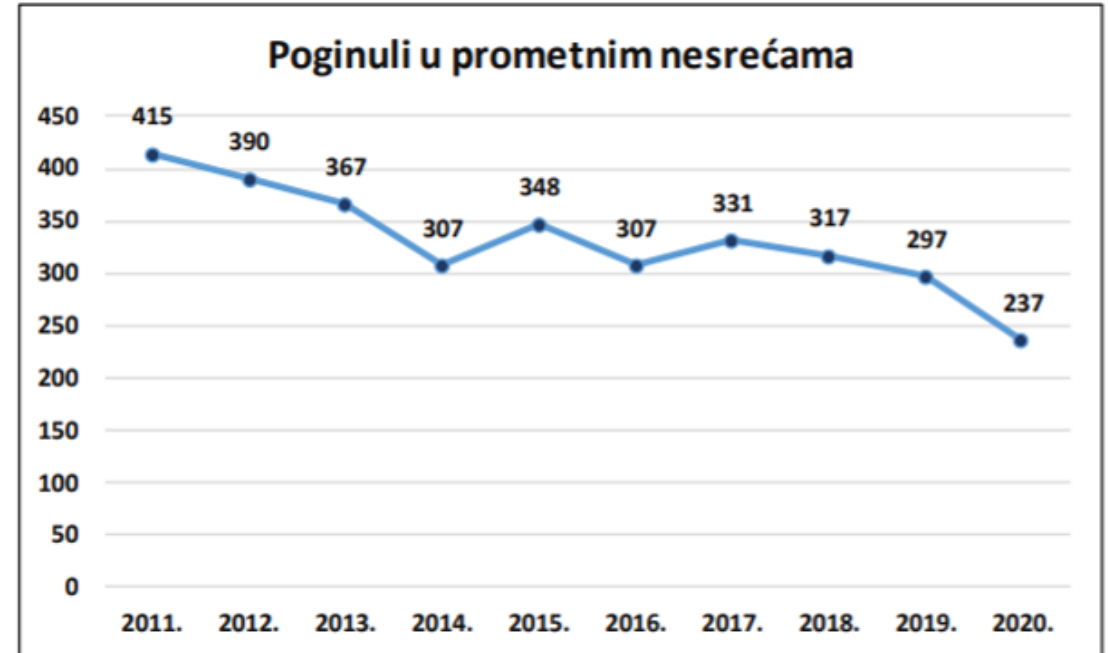
- </> XML
- </> JSON

Ključni podaci

- 📄 Država vozila
- 📄 Registarska oznaka
- 📄 Serijski broj uređaja ili geolokacija
- 📄 Brzina
- 📄 Prekoračenje brzine

Rezultati

- 30% manje prekršaja (negdje i do 50%)
- Smanjen broj teških nesreća zbog prebrze vožnje
- Automatiziran proces prijenosa i obrade snimaka
- Smanjeno vrijeme uručenja prekršaja
- Naplaćenim prekršajima vraćeno ulaganje



Provjera registarskih pločica

Radi se o servisu za provjeru registarskih pločica u stranim i nacionalnim evidencijama

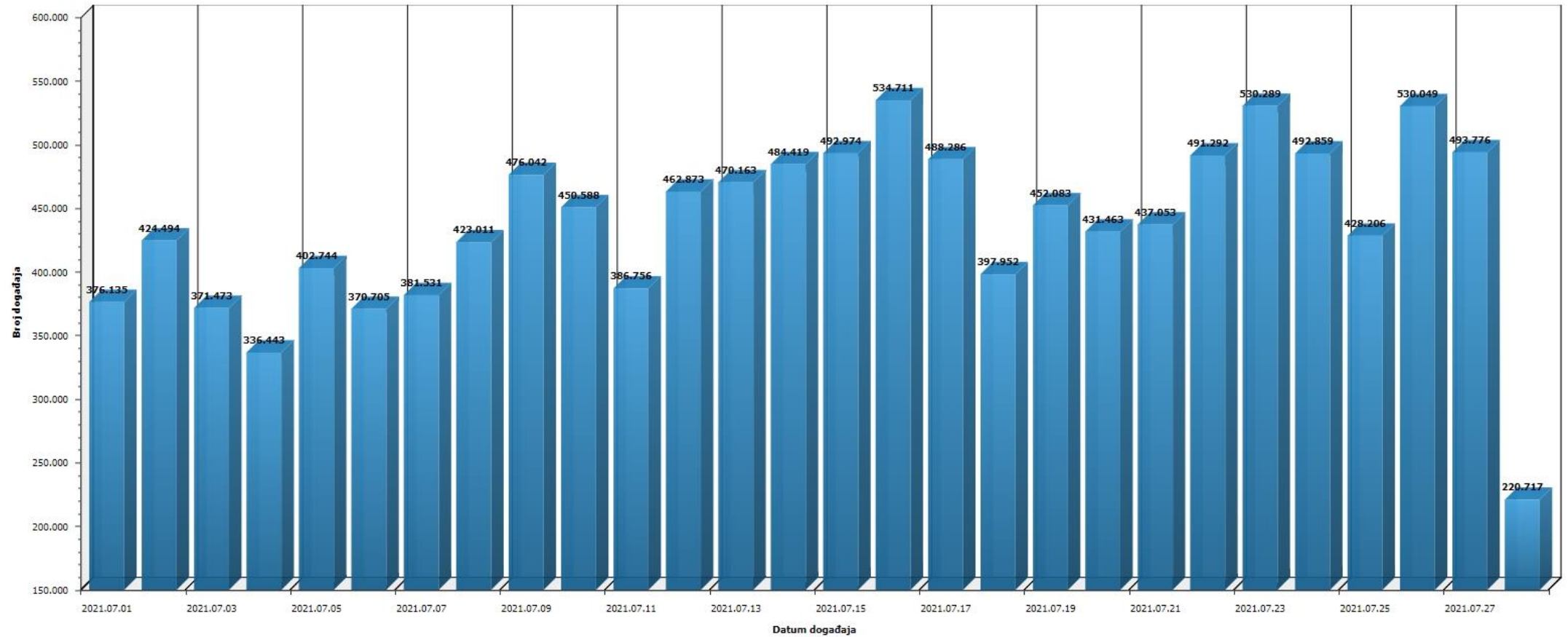
Korisnički zahtjev

- Integracija u realnom vremenu – trenutna dostupnost podataka
- Velika količina podataka – 50 upita po sekundi
- Planirana nadogradnja sustava – 150 upita po sekundi
- Dostupnost sustava 0 – 24
- MUP sigurnosna pravila – autentifikacija, autorizacija, logiranje
- Brza implementacija

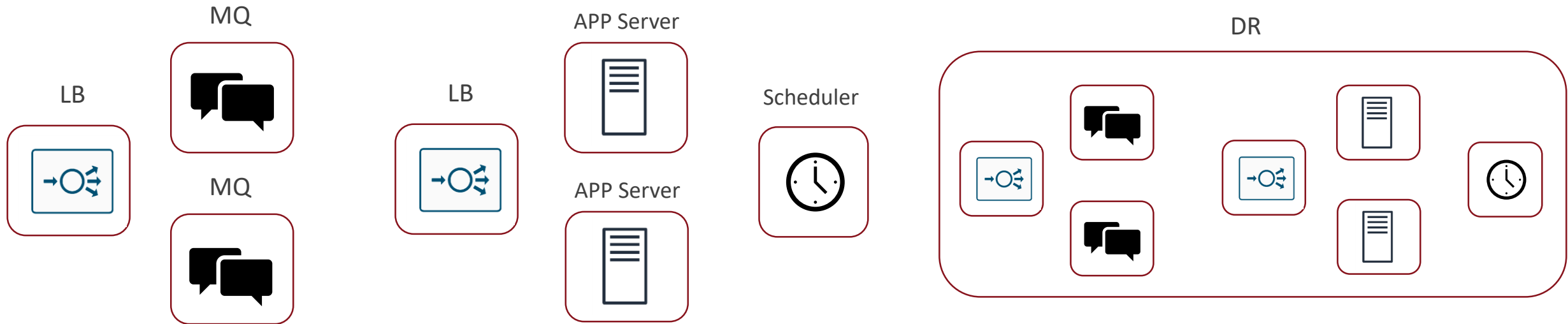


Korisnički zahtjev

Broj događaja po datumu



Arhitektura #1 - Overengineering



Elementi

- Message queues
- Redundantni aplikativni serveri
- Load balanceri
- Scheduleri
- Disaster recovery

Karakteristike

- Brzina – mogućnost obrade velike količine podataka
- Robusnost
- Sigurnost

Nedostaci

- Kašnjenje
- Složenost

„U strahu su velike oči”

Arhitektura #2 - Replikacija podataka

Elementi

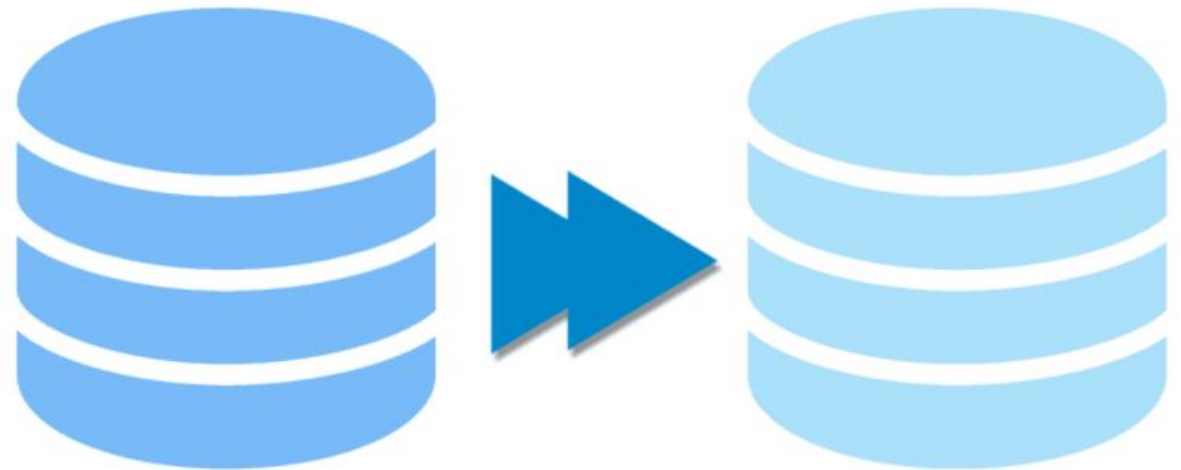
- Replikacija podataka
- Okidači (trigeri) – automatska sinkronizacija
- Poslovna logika na klijentskoj strani

Karakteristike

- Trenutna dostupnost podataka
- Maksimalna brzina obrade podataka
- Jednostavna arhitektura

Nedostaci

- Nestandardno / neindustrijsko rješenje
- Mogućnost neispravnih podataka



Arhitektura #2b - Industrijska replikacija podataka

Elementi

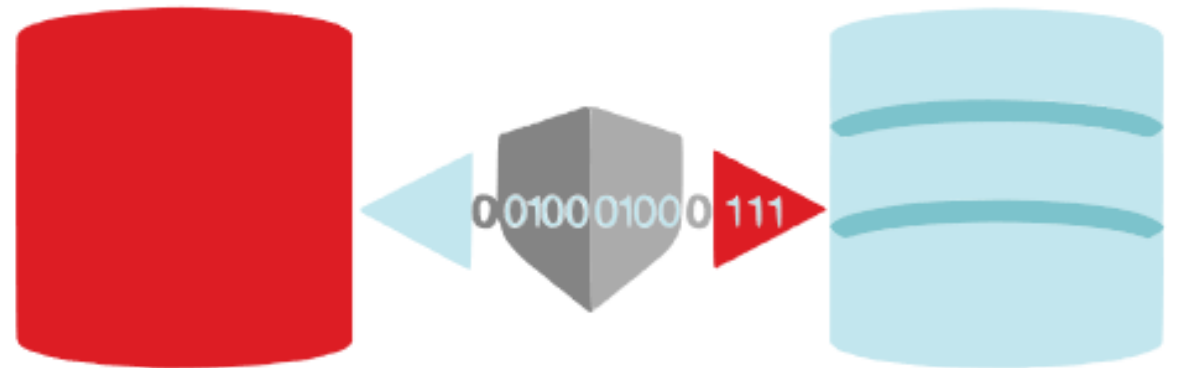
- Gotovo rješenje za replikaciju podataka – Oracle DataGuard
- Poslovna logika na klijentskoj strani

Karakteristike

- Industrijsko rješenje za replikaciju
- Trenutna dostupnost podataka
- Maksimalna brzina obrade podataka
- Maksimalna sigurnost podataka

Nedostaci

- Složenija arhitektura
- Dodatni zahtjevi za resursima, licencama i sl.



Arhitektura #3 - Web aplikacija

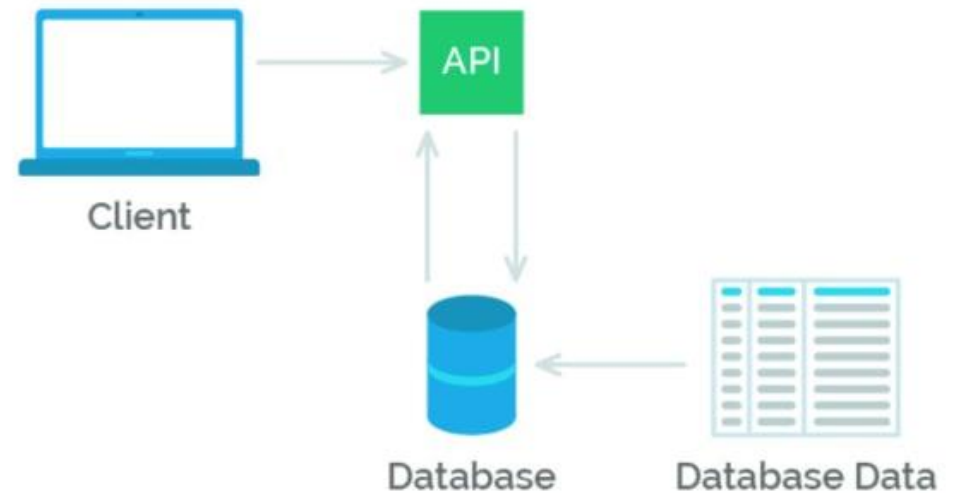
Elementi

- Spring Boot Java aplikacija
- Rest servisi
- Java tehnologije niske razine - JDBC
- Bazne procedure
- Optimizacija podataka - indeksi

Karakteristike

- Jednostavna implementacija
- Jednostavno održavanje
- Sinkrona obrada podataka
- Principi najbolje prakse

REST API Design



Nedostaci

- Potencijalno nedovoljno za tražene performanse

Stres test

Alati

- Apache JMeter
- Linux task manager programi – top i sl.
- Oracle monitoring - MUP informatika

Faze testiranja

- Test baze - bez aplikativnog dijela
- Test aplikacije na nižim razinama – razvoj, test
- Kratki testovi na produkcijskoj razini
- Cjelodnevni test u radno vrijeme na produkciji sa maksimalnim opterećenjem
- 24 satni test sa pravim podacima – test od strane OCR sustava

Stres test

```
top - 12:48:28 up 36 days, 21:59, 1 user, load average: 26.80, 24.97, 15.12
Tasks: 209 total, 2 running, 207 sleeping, 0 stopped, 0 zombie
%Cpu(s): 91.2 us, 0.1 sy, 0.0 ni, 8.6 id, 0.0 wa, 0.0 hi, 0.2 si, 0.0 st
KiB Mem : 32941056 total, 31207224 free, 1354068 used, 379764 buff/cache
KiB Swap: 8386556 total, 8386556 free, 0 used. 31448164 avail Mem
```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
61812	oracle	20	0	17.696g	1.011g	13204	S	1461	3.2	5132:38	java
1	root	20	0	54280	4080	2448	S	0.0	0.0	0:30.80	systemd
2	root	20	0	0	0	0	S	0.0	0.0	0:00.60	kthreadd
3	root	20	0	0	0	0	S	0.0	0.0	0:00.24	ksoftirqd/0
5	root	0	-20	0	0	0	S	0.0	0.0	0:00.00	kworker/0:0H
6	root	20	0	0	0	0	S	0.0	0.0	0:00.01	kworker/u:0
7	root	0	-20	0	0	0	S	0.0	0.0	0:00.00	kworker/u:0H
8	root	rt	0	0	0	0	S	0.0	0.0	0:00.16	migration/0
9	root	20	0	0	0	0	S	0.0	0.0	0:00.00	rcu_bh
10	root	20	0	0	0	0	R	0.0	0.0	0:48.17	rcu_sched
11	root	rt	0	0	0	0	S	0.0	0.0	0:09.56	watchdog/0
12	root	rt	0	0	0	0	S	0.0	0.0	0:08.28	watchdog/1
13	root	20	0	0	0	0	S	0.0	0.0	0:00.28	ksoftirqd/1
14	root	rt	0	0	0	0	S	0.0	0.0	0:00.12	migration/1
16	root	0	-20	0	0	0	S	0.0	0.0	0:00.00	kworker/1:0H
17	root	rt	0	0	0	0	S	0.0	0.0	0:08.41	watchdog/2
18	root	20	0	0	0	0	S	0.0	0.0	0:00.19	ksoftirqd/2
19	root	rt	0	0	0	0	S	0.0	0.0	0:00.15	migration/2
21	root	0	-20	0	0	0	S	0.0	0.0	0:00.00	kworker/2:0H
22	root	rt	0	0	0	0	S	0.0	0.0	0:08.20	watchdog/3

```
top - 12:48:29 up 36 days, 3:14, 1 user, load average: 28.88, 25.71, 15.31
Tasks: 210 total, 3 running, 207 sleeping, 0 stopped, 0 zombie
%Cpu(s): 91.9 us, 0.1 sy, 0.0 ni, 7.8 id, 0.0 wa, 0.0 hi, 0.2 si, 0.0 st
KiB Mem : 32941056 total, 31269080 free, 1323880 used, 348096 buff/cache
KiB Swap: 8386556 total, 8386556 free, 0 used. 31478838 avail Mem
```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
60688	oracle	20	0	17.700g	0.981g	13244	S	1473	3.1	5206:12	java
62649	root	20	0	130012	1884	1276	R	0.3	0.0	24:38.81	top
1	root	20	0	54276	4052	2436	S	0.0	0.0	0:32.39	systemd
2	root	20	0	0	0	0	S	0.0	0.0	0:00.69	kthreadd
3	root	20	0	0	0	0	S	0.0	0.0	0:00.16	ksoftirqd/0
5	root	0	-20	0	0	0	S	0.0	0.0	0:00.00	kworker/0:+
6	root	20	0	0	0	0	S	0.0	0.0	0:00.01	kworker/u:0
7	root	0	-20	0	0	0	S	0.0	0.0	0:00.00	kworker/u:+
8	root	rt	0	0	0	0	S	0.0	0.0	0:00.17	migration/0
9	root	20	0	0	0	0	S	0.0	0.0	0:00.00	rcu_bh
10	root	20	0	0	0	0	R	0.0	0.0	0:55.11	rcu_sched
11	root	rt	0	0	0	0	S	0.0	0.0	0:10.12	watchdog/0
12	root	rt	0	0	0	0	S	0.0	0.0	0:08.90	watchdog/1
13	root	20	0	0	0	0	S	0.0	0.0	0:00.18	ksoftirqd/1
14	root	rt	0	0	0	0	S	0.0	0.0	0:00.20	migration/1
16	root	0	-20	0	0	0	S	0.0	0.0	0:00.00	kworker/1:+
17	root	rt	0	0	0	0	S	0.0	0.0	0:08.68	watchdog/2
18	root	20	0	0	0	0	S	0.0	0.0	0:00.25	ksoftirqd/2
19	root	rt	0	0	0	0	S	0.0	0.0	0:00.18	migration/2
21	root	0	-20	0	0	0	S	0.0	0.0	0:00.00	kworker/2:+

```
top - 12:25:00 up 28 days, 21:36, 1 user, load average: 0.00, 0.02, 0.05
Tasks: 210 total, 1 running, 209 sleeping, 0 stopped, 0 zombie
%Cpu(s): 0.0 us, 0.0 sy, 0.0 ni, 100.0 id, 0.0 wa, 0.0 hi, 0.0 si, 0.0 st
KiB Mem : 32941056 total, 31393964 free, 1184028 used, 363064 buff/cache
KiB Swap: 8386556 total, 8386556 free, 0 used. 31630610 avail Mem
```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
55962	root	20	0	130012	1888	1264	R	0.7	0.0	0:00.46	top
397	root	20	0	0	0	0	S	0.3	0.0	0:03.05	xfsaild/sda3
1	root	20	0	54280	4076	2448	S	0.0	0.0	0:25.10	systemd

```
top - 12:25:00 up 28 days, 2:50, 1 user, load average: 106.53, 101.83, 71.10
Tasks: 214 total, 2 running, 212 sleeping, 0 stopped, 0 zombie
%Cpu(s): 98.6 us, 0.1 sy, 0.0 ni, 1.2 id, 0.0 wa, 0.0 hi, 0.1 si, 0.0 st
KiB Mem : 32941056 total, 31529768 free, 1076848 used, 334440 buff/cache
KiB Swap: 8386556 total, 8386556 free, 0 used. 31733604 avail Mem
```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
52156	oracle	20	0	17.817g	763284	13232	S	1580	2.3	429:19.59	java
55156	root	20	0	130012	1864	1264	R	0.3	0.0	0:00.34	top
1	root	20	0	54276	4052	2436	S	0.0	0.0	0:26.39	systemd

Rezultati testiranja

Aggregate Report


Name:

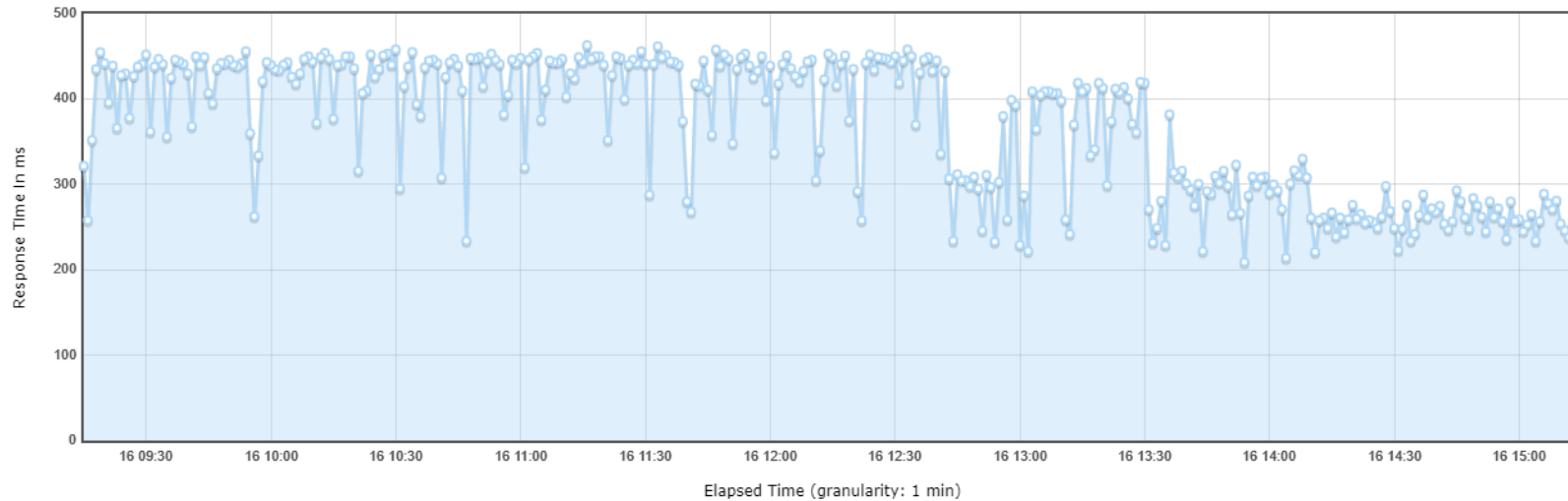
Comments:

Write results to file / Read from file

Filename: Log/Display Only: Errors Successes

Label	# Samples	Average	Median	90% Line	95% Line	99% Line	Min	Maximum	Error %	Throughput	Received KB/sec	Sent KB/sec
anpr/check/resp...	2344881	249	211	404	442	501	120	3348	0,01%	108,6/sec	47,39	34,79
TOTAL	2344881	249	211	404	442	501	120	3348	0,01%	108,6/sec	47,39	34,79

 Response Time Percentiles Over Time (successful responses)



90th percentile 95th percentile 99th percentile Max Median Min

Rezultati testiranja

Rezultati stres testa

- Više od 200 upita unutar jedne sekunde
- prosječno vrijeme za 90% upita 0,4 sekunde
- Ukupan broj poslanih upita sa jednog računala > od 2 milijuna kroz 6h

Rezultati predprodukcije

- 5 – 25 upita u sekundi
- Više od 500 000 upita
- 700-tinjak pronađenih vozila

Zaključak

- ✓ Ponekad je jednostavno rješenje najbolje
- ✓ Spring boot – spreman za produkciju sa standardnim postavkama
- ✓ Slušajte svoje korisnike



Hvala na pažnji!

Buzinski prilaz 10, 10010 Zagreb, Hrvatska
+385 (0)1 6690 800
prodaja@king-ict.hr
www.king-ict.hr

