



—

# Osnove Performance Testiranja

**KURS OSNOVE TESTIRANJA SOFTVERA**

**STEFAN MILADINOVIĆ – SENIOR TESTER**  
**BORIS BOŠKOV – SENIOR TESTER**



# Osnove performance testiranja

- 1. DEFINICIJA PERFORMANSI**
- 2. PROCES PERFORMANCE TESTIRANJA**
- 3. TIPOVI PERFORMANCE TESTIRANJA**
  
- 4. PRIMER JMETER PERFORMANCE SKRIPTE**
- 5. PRIMER IZVEŠTAJA PERFORMANCE TESTA**



OSNOVE PERFORMANCE TESTIRANJA

# Definicija performansi





## Primer loših performansi

Planinar je u planini, planinari.

Želi da kupi karte za koncert veoma popularnog benda.

Odabere dve karte, stavi ih u korpu i pređe na plaćanje. Sistem insistira da se uloguje, ali nema nalog.

Pravljenje naloga traje predugo i ima previše koraka.

Konačno uspeva da dođe do plaćanja.

Plaćanje traje preko 30 sekundi, nakon čega dobio 500 Internal Server Error.

Dobija SMS od banke da je uplata prošla.

Karte ne stižu na mejl.

Planinar je veoma ljut, i neće više kupovati od ove firme.



## Primer loših performansi: Šta su sve problemi?

- Sistem očekuje od korisnika da budu ulogovani

*Potreba za logovanjem pravi nepotrebno opterećenje*

- Kreiranje naloga traje predugo

*Mogući problemi sa bazom podataka?*

- Novac je prebačen, ali roba nije isporučena

*Loše rešena obrada grešaka*

*Rezultat:* puno nezadovoljnih korisnika -> negativan uticaj na imidž kompanije -> negativan uticaj na poslovanje -> negativan uticaj na zaradu

# Kako definišemo dobre performanse?

Pogledajmo vreme potrebno da aplikacija odreaguje na komandu:

- ▶ **Više od 15 sekundi**

Samo pojedini korisnici mogu tolerisati ovo odlaganje. Mora se obezbediti nastavak rada na drugom procesu

- ▶ **Više od 4 sekunde**

Ovo odlaganje je predugačko za unos podataka ili rešavanje problema, ali je prihvatljivo za jednostavnije obrade podataka ili računanje

- ▶ **2 do 4 sekunde**

Za neke složenije primene, poput plaćanja, ovo je prihvatljivo. Za pregled onlajn kataloga - nije

- ▶ **Manje od 2 sekunde**

Granično vreme za poslove koji imaju više koraka i zahtevaju pamćenje prethodnih koraka

- ▶ **Kraće od 1 sekunde**

Gornja granica za kreativne poslove

- ▶ **0.1 sekunda i brže**

Pomeranje i klik mišem, kucanje na tastaturi

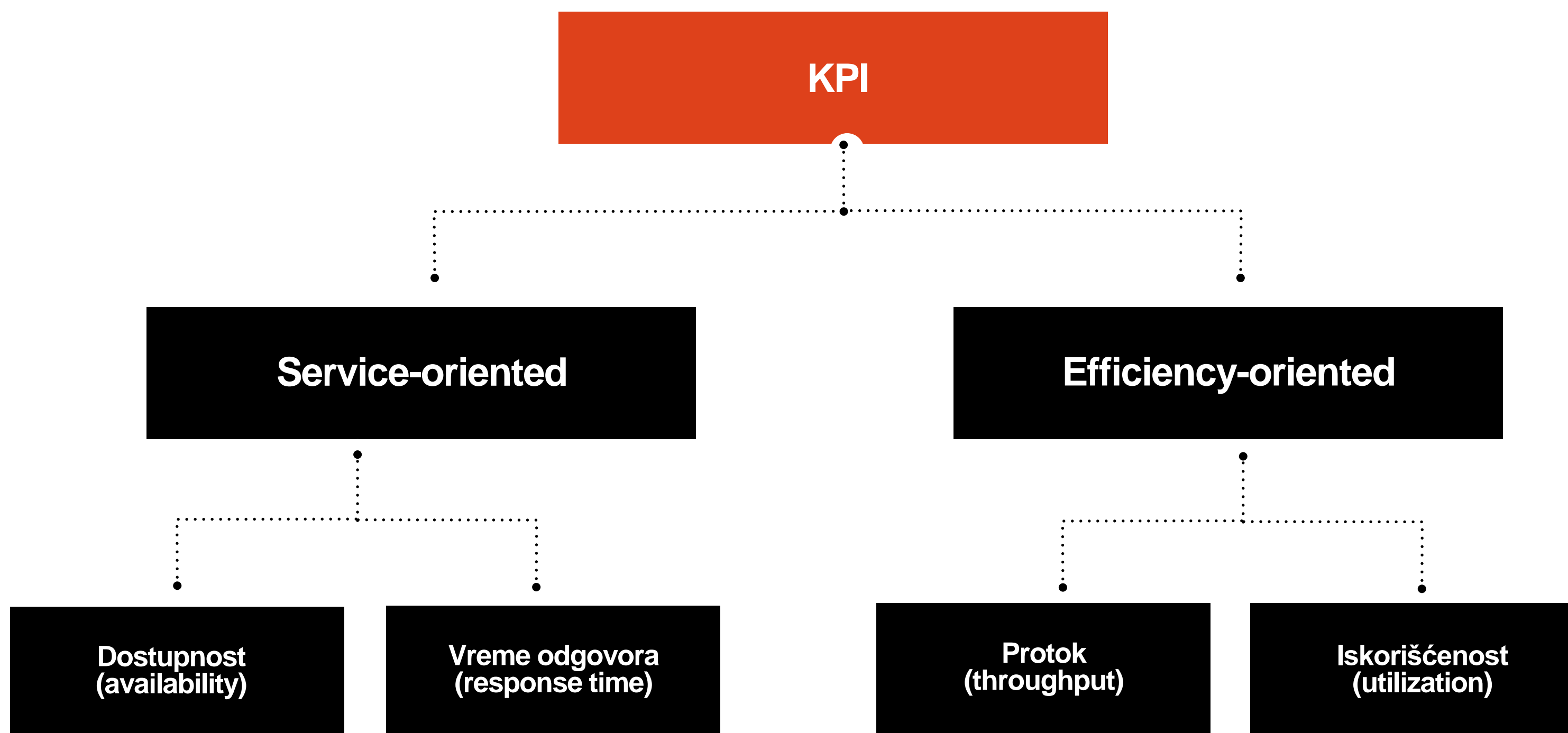
# Kako definišemo dobre performanse?

Bitan zaključak:

***Ne postoji univerzalna vrednost koja će reći da li softver ima dobre ili loše performanse.***

***Performanse softvera jako zavise od konteksta upotrebe.***

Kako onda ispitujemo da li softver ima dobre performanse?



Koliko dobro aplikacija pruža uslugu korisnicima?

Koliko dobro aplikacija iskorišćava resurse koji su joj na raspolaganju?

Key Performance Indicators

KPI



# Key Performance Indicators

- ▶ Dostupnost - Availability

*Vreme tokom kojeg je aplikacija dostupna korisnicima*

- ▶ Vreme odgovora - Response Time

*Vreme potrebno da aplikacija odreaguje na zahtev korisnika*

- ▶ Protok - Throughput

*Broj značajnih događaja u jedinici vremena*

- ▶ Iskorišćenost - Utilization

*Procenat teorijskog kapaciteta resursa koji aplikacija koristi*



OSNOVE PERFORMANCE TESTIRANJA

# Proces Performance testiranja



# Priprema

*Test Planning – odabir alata i test strategije*

*Test Analysis and Design – performance zahtevi koje aplikacija mora ispuniti*



*Merljivi parametri i njihove očekivane vrednosti*

*Test Implementation – kreiranje podataka i okruženja za test*



*Test skripte u odabranom alatu*  
*Podignuto test okruženje*

## Izvršavanje i kraj

***Test Execution – set rezultata***



*Obrada rezultata i poređenje sa zahtevima. Ponavljanje po potrebi.  
Pronalaženje uzroka u slučaju problema.*

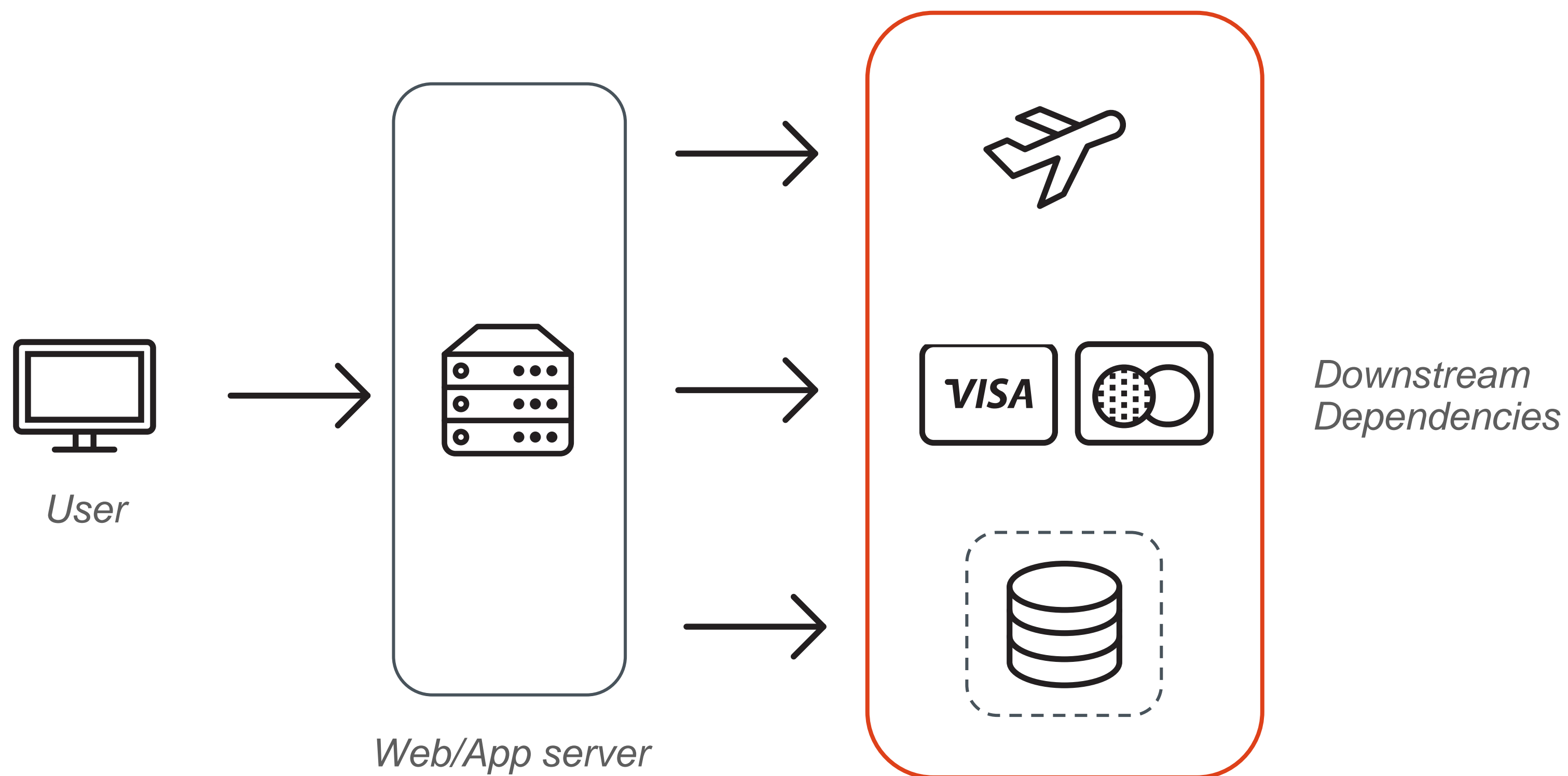
***Test Reporting – kreiranje izveštaja***



*Izveštaji koji su prilagođeni publici.*

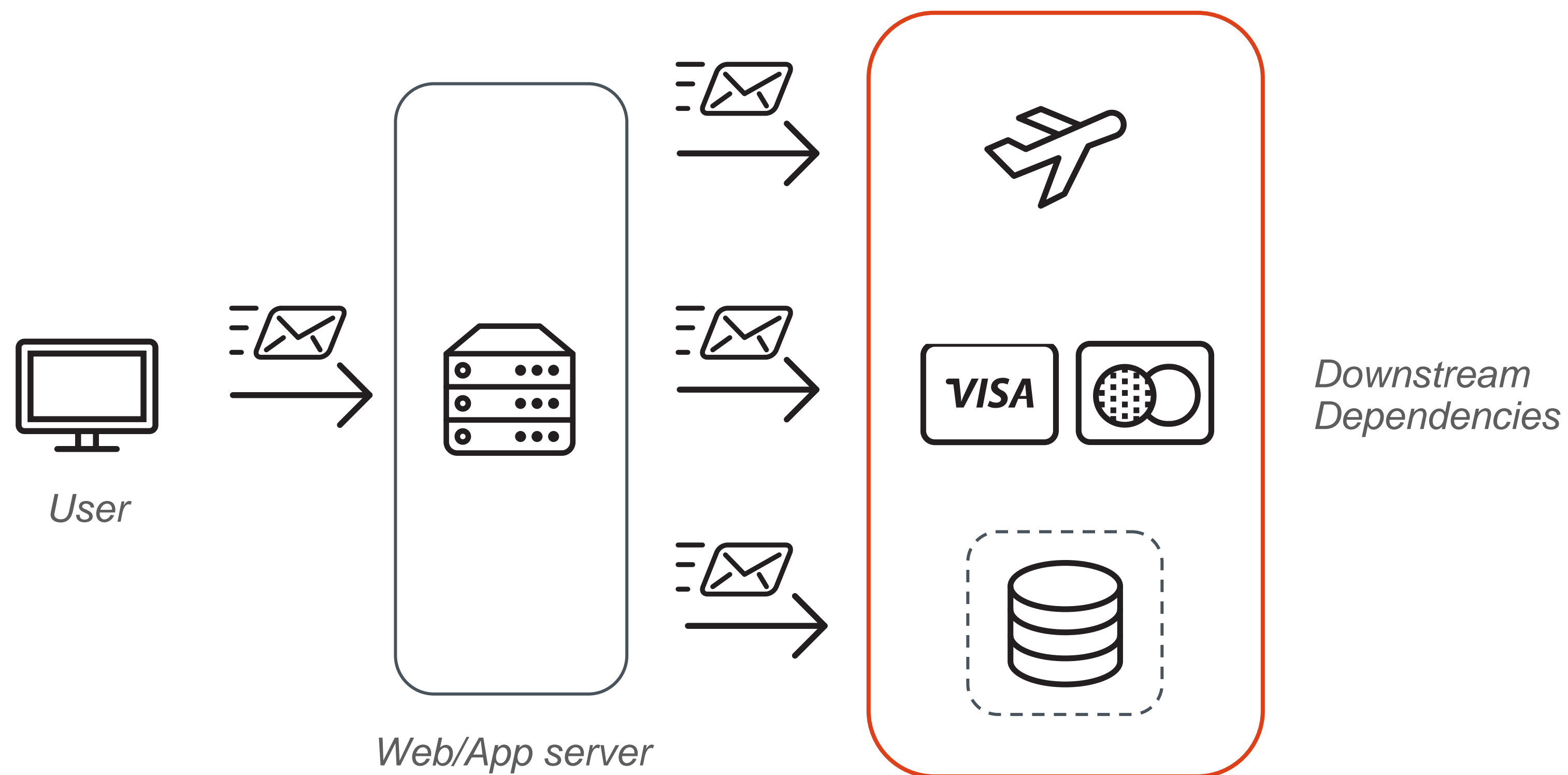
# Primer infrastrukture za Performance test

*Primer: sistem za online kupovinu avionskih karata – produkcijski sistem*



# Primer infrastrukture za Performance test

*Primer: sistem za online kupovinu avionskih karata – produkcijski sistem*



# HTTP poruke

Šta su HTTP poruke?

Kome se šalje poruka?

Kako se šalje poruka?

Šta sadrži poruka?



Zahtev

https://10.47.17.139:8443/api/rest/rprm/endpoint-groups/

POST

```
{  
  "endpoint-count": 0,  
  "endpoint-group-name": "test",  
  "endpoint-summary-list": {  
    "plcm-epg-endpoint-summary": [  
      ]  
    },  
  "profile-summary-list": {  
    "plcm-profile-summary": [  
      ]  
    }  
  }  
}
```



Odgovor



Status poruke 201

Odgovor

```
{  
  "anies": null,  
  "business-types": null,  
  "profile-summary-list": {  
    "anies": null,  
    "entity-tag": "d41d8cd98f00b204e9800998ecf8427e",  
    "plcm-profile-summary-v2": null  
  },  
  "related-id": "8a960d0b-6e16-4190-b2a9-ee5ac321307",  
  "related-name": "test"  
}
```

# HTTP poruke

Šta su HTTP poruke?



Address

https://10.47.17.139:8443/api/rest/rprm/endpoint-groups/

Method

POST

Request Body

```
{  
  "endpoint-count": 0,  
  "endpoint-group-name": "test",  
  "endpoint-summary-list": {  
    "plcm-epg-endpoint-summary": [  
      ]  
    },  
  "profile-summary-list": {  
    "plcm-profile-summary": [  
      ]  
    }  
}
```



Zahtev

Odgovor



Status/Response Code

201

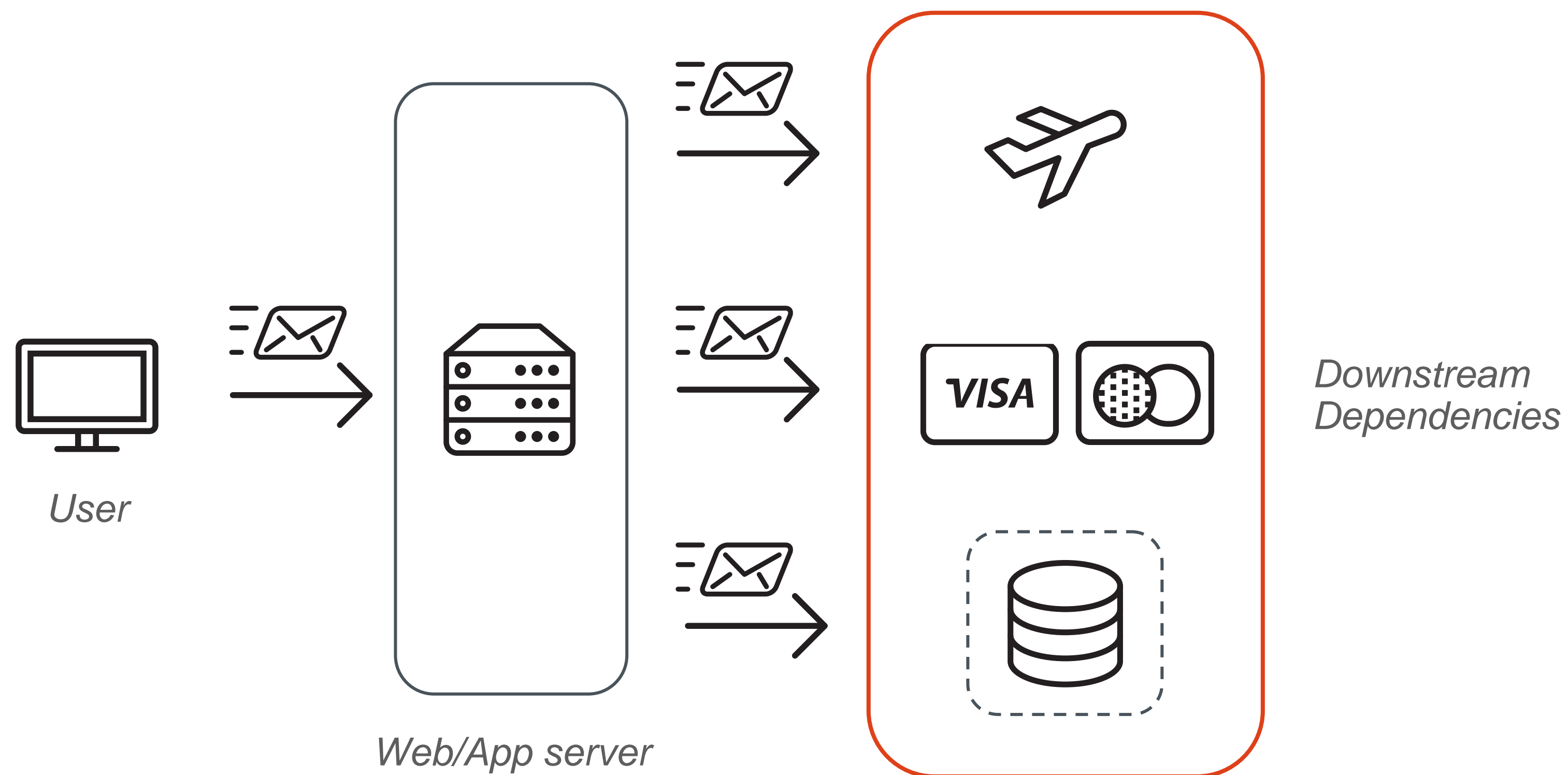
Response Body

```
{  
  "anies": null,  
  "business-types": null,  
  "profile-summary-list": {  
    "anies": null,  
    "entity-tag": "d41d8cd98f00b204e9800998ecf8427e",  
    "plcm-profile-summary-v2": null  
  },  
  "related-id": "8a960d0b-6e16-4190-b2a9-ee5ac321307",  
  "related-name": "test"  
}
```



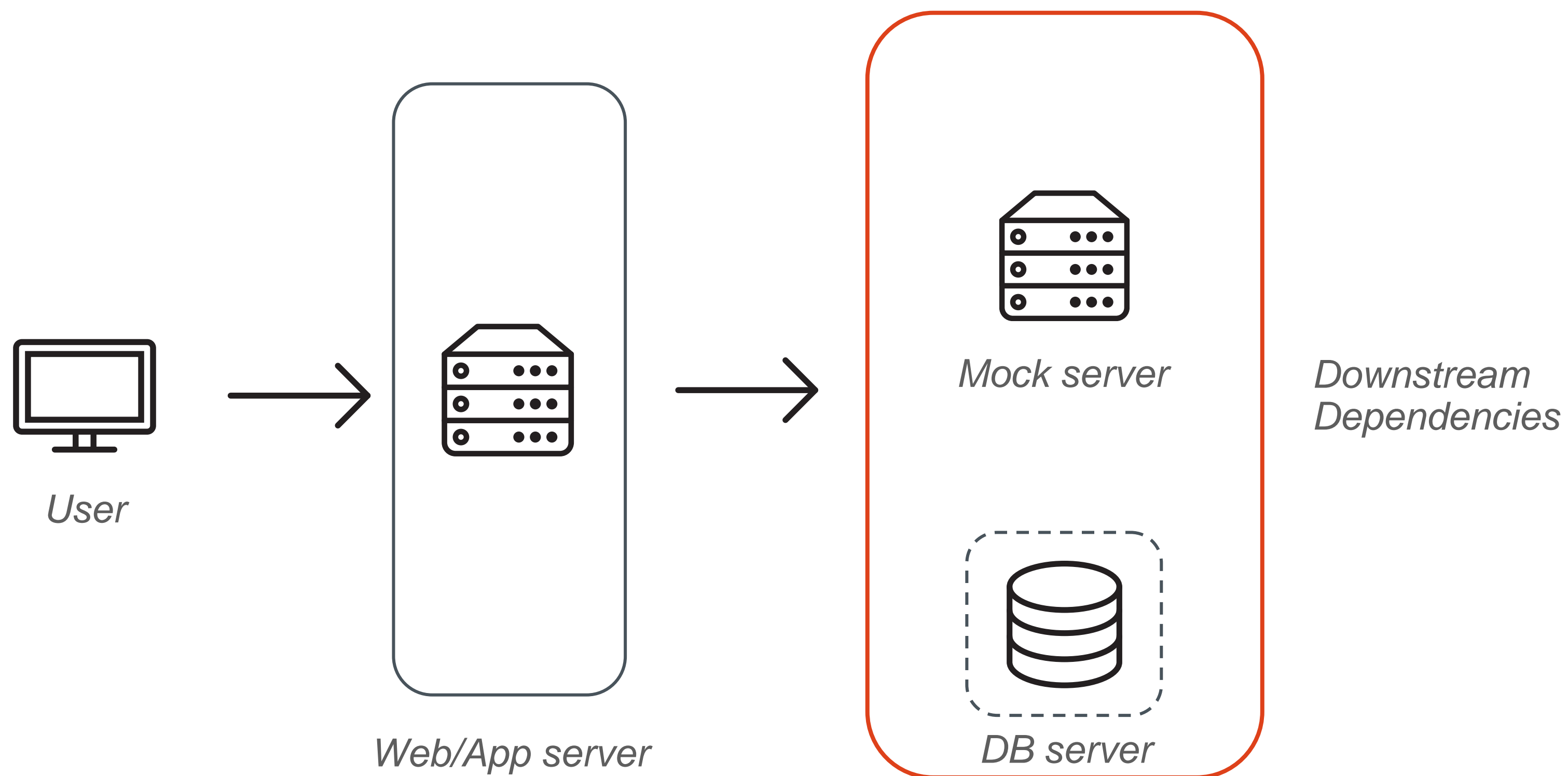
# Primer infrastrukture za Performance test

*Primer: sistem za online kupovinu avionskih karata – produkcijski sistem*



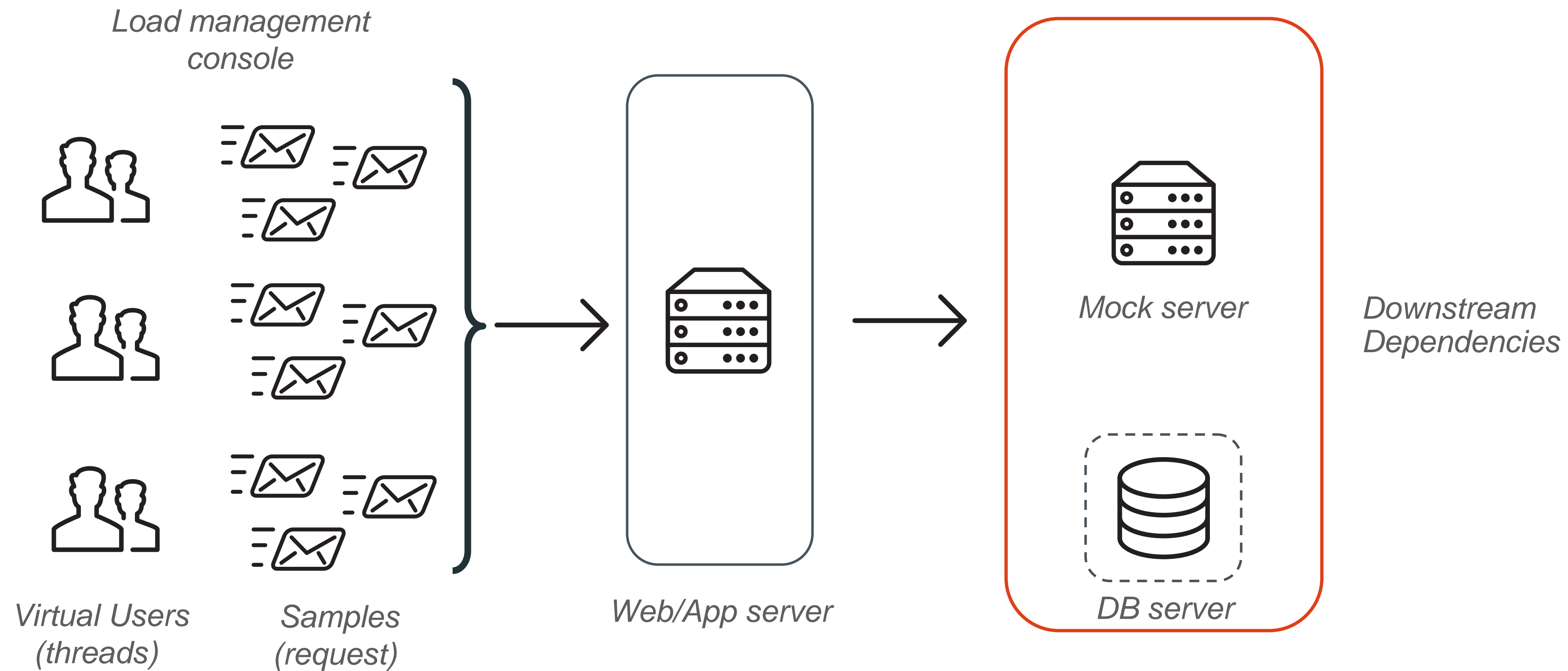
# Primer infrastrukture za Performance test

*Primer: sistem za online kupovinu avionskih karata – performance testing sistem*



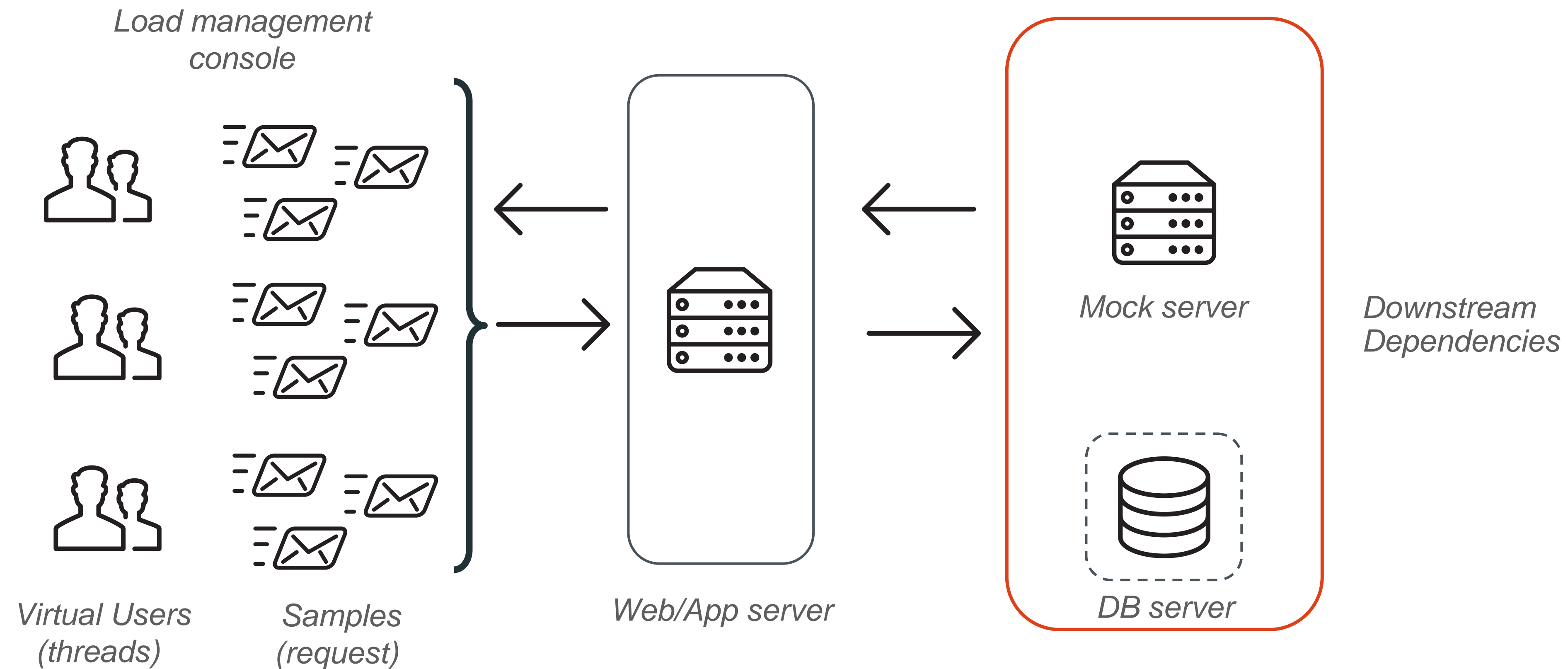
# Primer infrastrukture za Performance test

*Primer: sistem za online kupovinu avionskih karata – performance testing sistem*



# Primer infrastrukture za Performance test

*Primer: sistem za online kupovinu avionskih karata – performance testing sistem*





OSNOVE PERFORMANCE TESTIRANJA

# Tipovi Performance testiranja



# Tipovi Performance testiranja

Prema ISTQB

- ▶ Load
- ▶ Spike
- ▶ Endurance
- ▶ Stress
- ▶ Concurrency

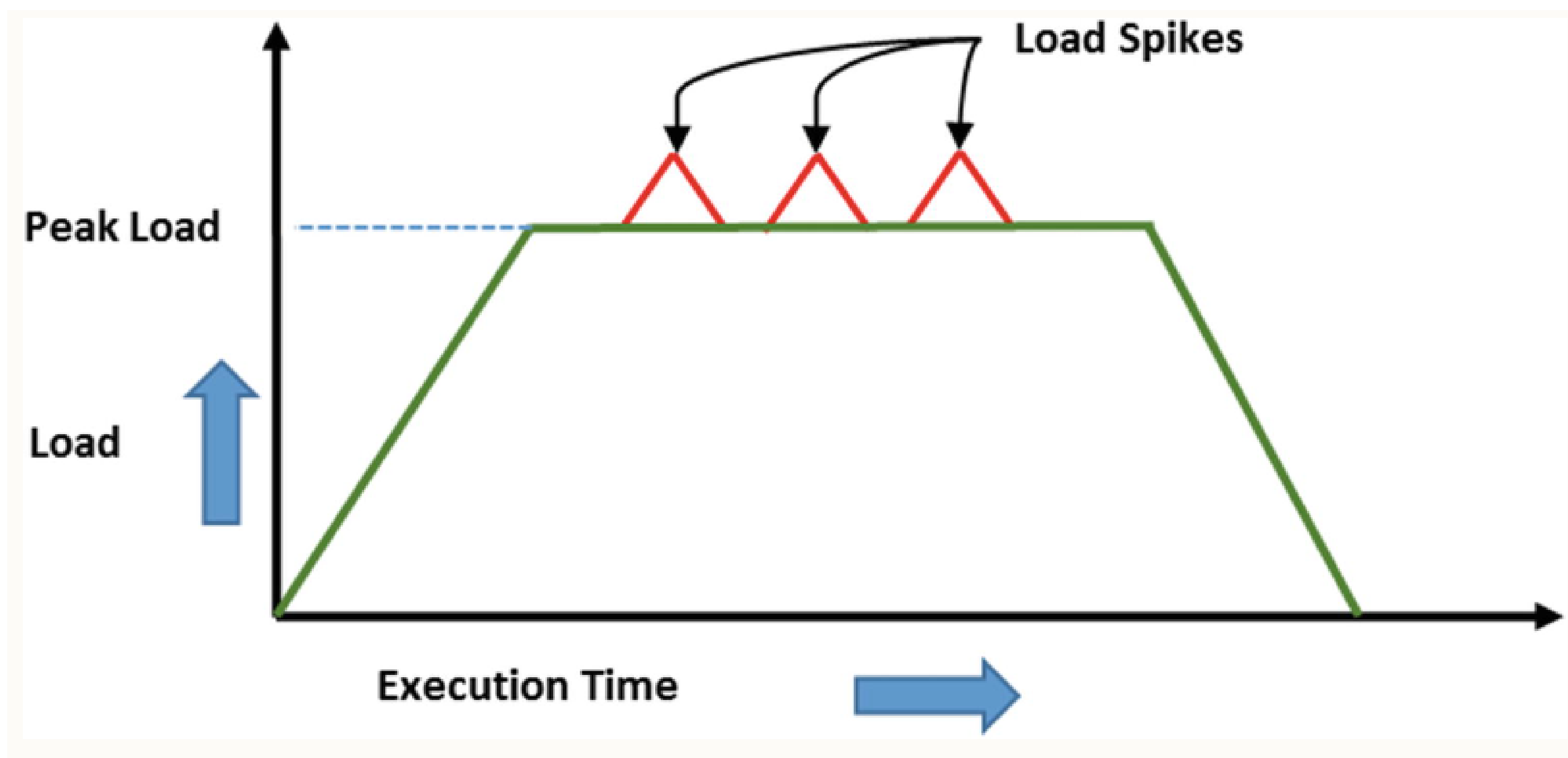
# Load testing

- *Opterećenje: standardno projektovano opterećenje koje aplikacija mora izdržati*
- *Trajanje: relativno kratko, nekoliko sati*
- *Osnova za dalja Performance testiranja*

Ispituje ponašanje aplikacije pod očekivanim opterećenjem, i to:

- niskim
- tipičnim
- visokim

# Spike testing



*PERFORMANCE TESTING: AN ISTQB CERTIFIED TESTER FOUNDATION LEVEL SPECIALIST CERTIFICATION REVIEW – KEITH YORKSTON*

Ispituje kako aplikacija reaguje na kratkotrajna preopterećenja.



# Endurance (Soak) testing

- *Opterećenje: kao i za Load test → standardno*
- *Trajanje: veoma dugo (dani, ponekad i nedelje)*
- *Mogući problemi: memory leaks, preveliki logovi, thread pool-ovi potrošeni*
- *Mana: nema uvek dovoljno vremena za Soak test*

Ispituje ponašanje aplikacije pri veoma dugom standardnom opterećenju.

# Stress testing

- *Tip opterećenja*

Mnogo veće opterećenje od standardnog, treba da izazove preopterećenje

- *Trajanje*

Zavisi od ponašanja aplikacije, ali ne duže od nekoliko sati

- *Koristi se za*

Testiranje disaster scenarija, ispitivanje stabilnosti aplikacije, simuliranje Denial of Service napada

*Napomena: bitno je pripremiti sistem pre početka, opterećivanjem standardnim opterećenjem. Takođe je bitno ispit*

**Ispituje ponašanje aplikacije pod abnormalno visokim opterećenjem**

**Cilj: pronaći tačku pucanja aplikacije**

# Stress testing

## ***Bitno!***

*Pripremiti sistem puštanjem standardnog Load testa, i pustiti isto opterećenje na kraju testa*

## ***Veoma bitno!***

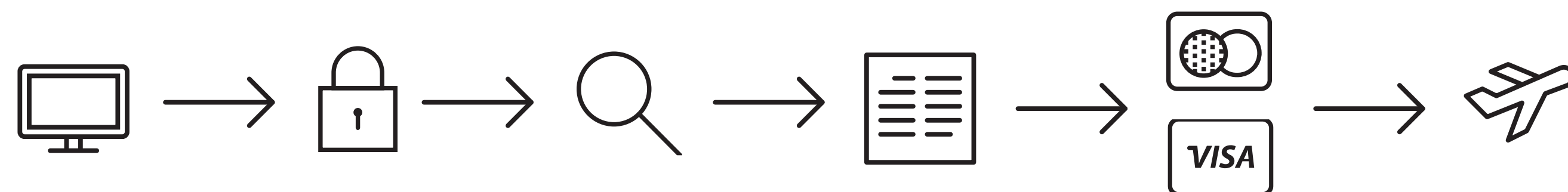
*Šta se dešava sa aplikacijom nakon prestanka opterećenja?*

**Ispituje ponašanje aplikacije pod abnormalno visokim opterećenjem**

**Cilj: pronaći tačku pucanja aplikacije**

# Concurrency testing

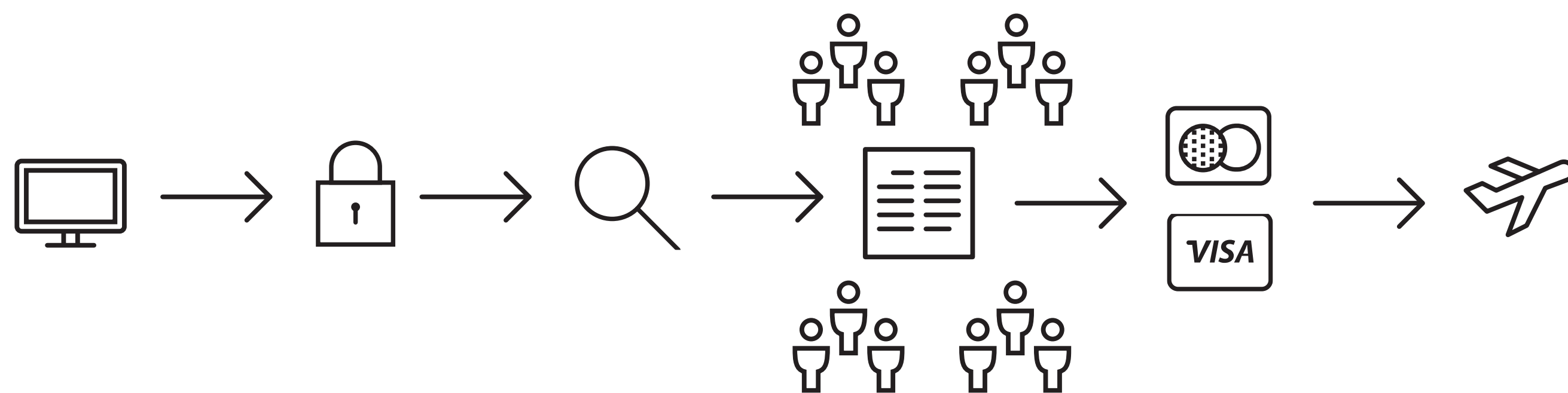
*Primer: sistem za online kupovinu karata*



Ispituje ponašanje aplikacije pri povećanom opterećenju pojedinih segmenata.

# Concurrency testing

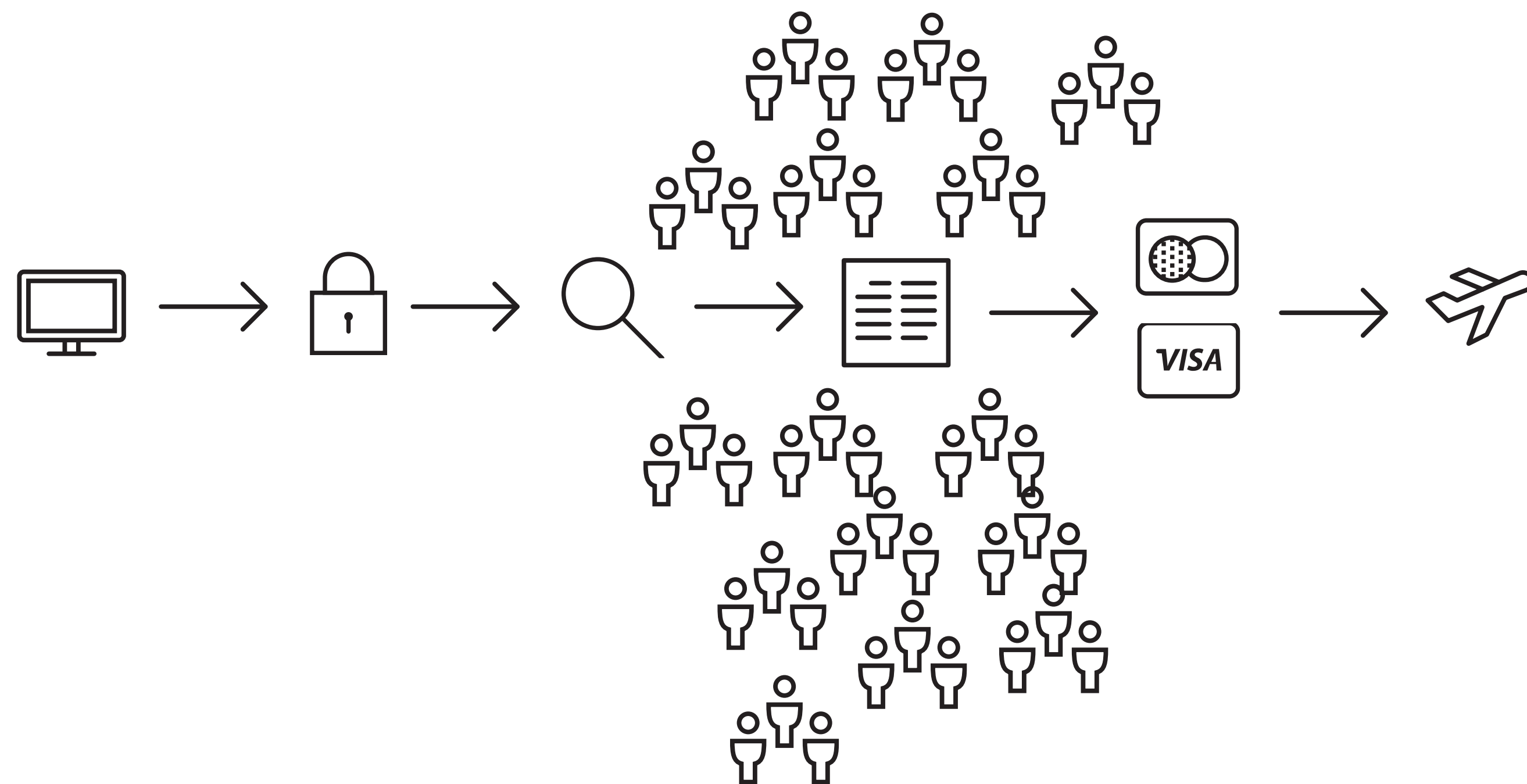
*Primer: sistem za online kupovinu karata*



Ispituje ponašanje aplikacije pri povećanom opterećenju pojedinih segmenata.

# Concurrency (isolation) testing

*Primer: sistem za online kupovinu karata*



Ispituje ponašanje aplikacije pri povećanom opterećenju pojedinih segmenata.

## Završne reči

***Performance testiranje je tehnika nefunkcionalnog testiranja softvera – ne proverava samo šta softver radi, već i kako to radi.***

***Ne postoji univerzalni set veličina i njihovih vrednosti koji određuju dobre (ili loše) performanse svakog softvera.***

***Performance testiranje je veoma blisko eksperimentu – pripremamo okruženje, vršimo merenje, analiziramo rezultate.***

***Dobro poznavanje poslovnog procesa i aplikacije je neophodno!***