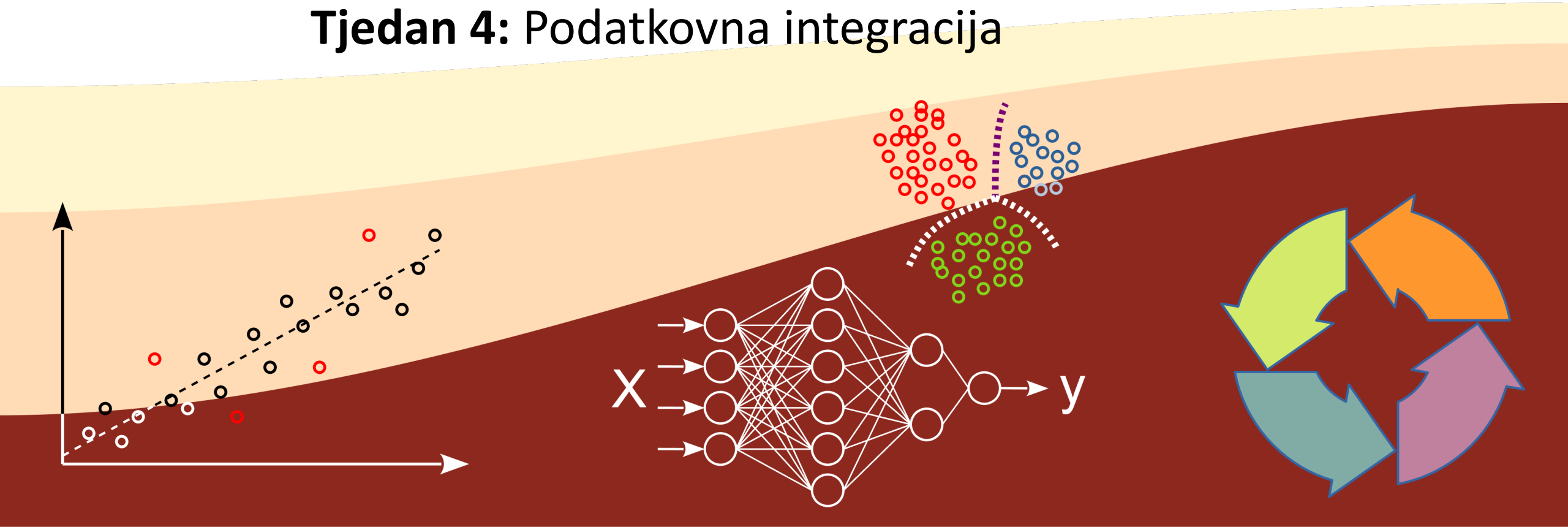
Arhitektura i Razvoj Inteligentnih Sustava

Tjedan 4: Podatkovna integracija



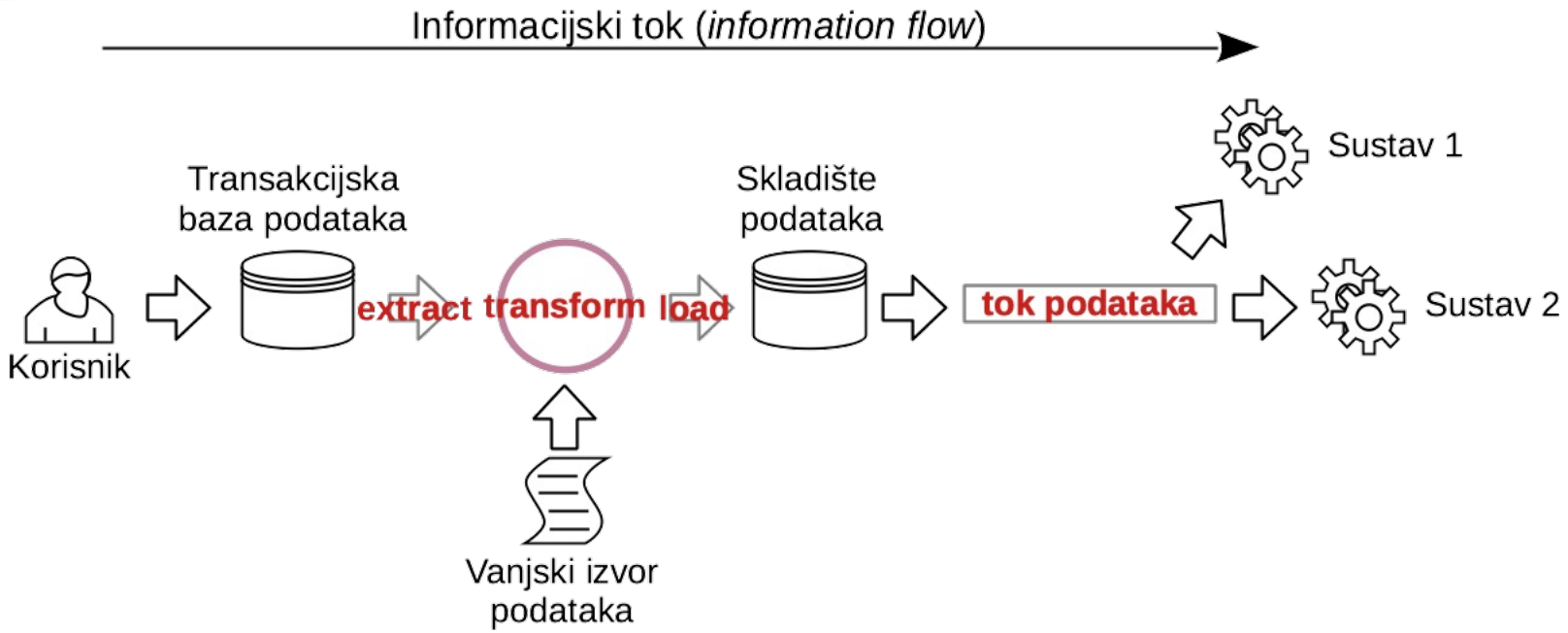
Creative Commons

- slobodno smijete:
  - dijeliti — umnožavati, distribuirati i javnosti priopćavati djelo
  - prerađivati djelo
- pod sljedećim uvjetima:
  - imenovanje: morate priznati i označiti autorstvo djela na način kako je specificirao autor ili davatelj licence (ali ne način koji bi sugerirao da Vi ili Vaše korištenje njegova djela imate njegovu izravnu podršku).
  - nekomercijalno: ovo djelo ne smijete koristiti u komercijalne svrhe.
  - dijeli pod istim uvjetima: ako ovo djelo izmijenite, preoblikujete ili stvarate koristeći ga, preradu možete distribuirati samo pod licencom koja je ista ili slična ovoj.

Informacijski tokovi (*information flow*)

- Apstraktni poslovni koncept
  - Podaci imaju svoje izvore: korisnici (klijenti), vanjske organizacije, senzori
  - Organiziramo i spremamo ih u podatkovna spremišta
    - Permanentna
      - Relacijske baze podataka, skladišta podataka
      - Specijalizirane baze podataka, nerelacijske baze podataka, NoSQL
      - Datoteke, Key-Value spremišta podataka
    - Tranzitivna
      - Tokovi podataka (*stream*) – Apache Kafka
  - Informacije (podaci) se kroz kompleksni proces i integraciju sustava i komponenti prenose od svog izvorišta do mjesta konzumacije
    - U međuvremenu imamo razne transformacije i agregacije podataka
    - Podaci se mogu dodatno specificirati i oplemeniti kroz dodatne interakcije sa korisnicima

Informacijski tokovi



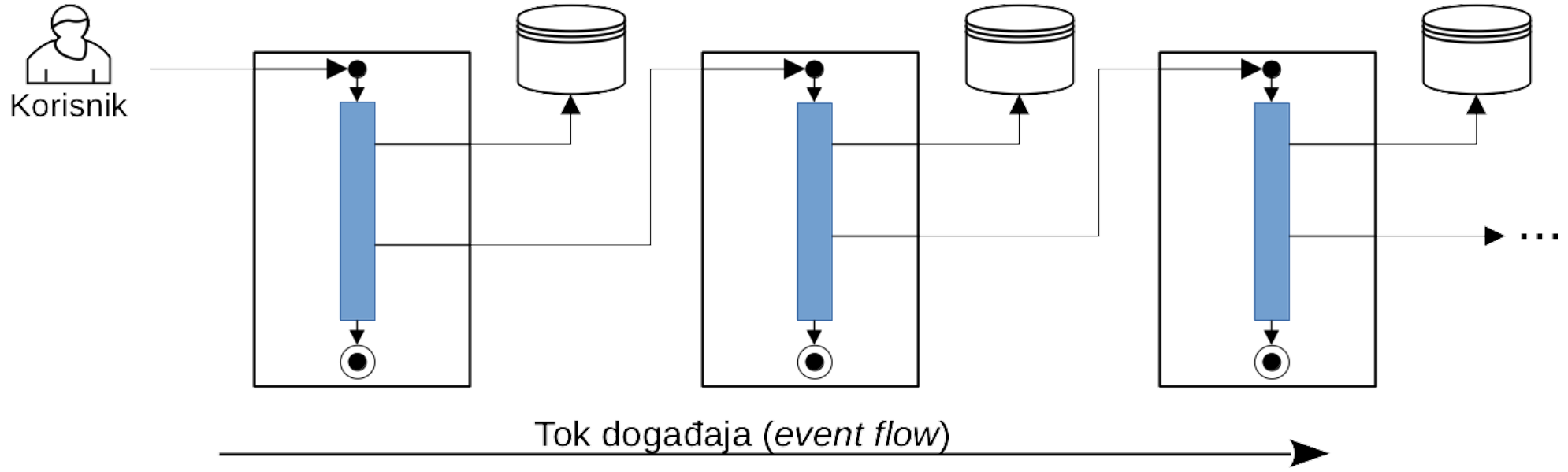
- Podatkovni cjevovod (*data pipeline*)
- Kako uopće implementirati ovakav podatkovni cjevovod?
  - Već smo spominjati kompleksni proces, kao i integracije
  - Želimo automatizirati ovakav cjevovod

Za automatizaciju

- Moramo imati jedno od:
  - Prijenos događaja od integracije do integracije
  - Pomoćni sustav koji omogućava automatizaciju podatkovnog cjevovoda
- Prijenos događaja (*event-driven*)
  - Kompleksan
  - Jako ovisi o sustavima i komponentama koje su u cjevovodu
  - Prijenos događaja može biti prekinut kada se u cjevovodu u slijedu nađu dva sustava koji ne iniciraju komunikaciju
    - na primjer dvije baze podataka

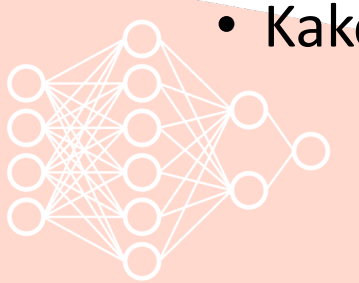
Samopodržavajući cjevovod

- Postoji slijed sustava koji prenose događaje jedan drugome

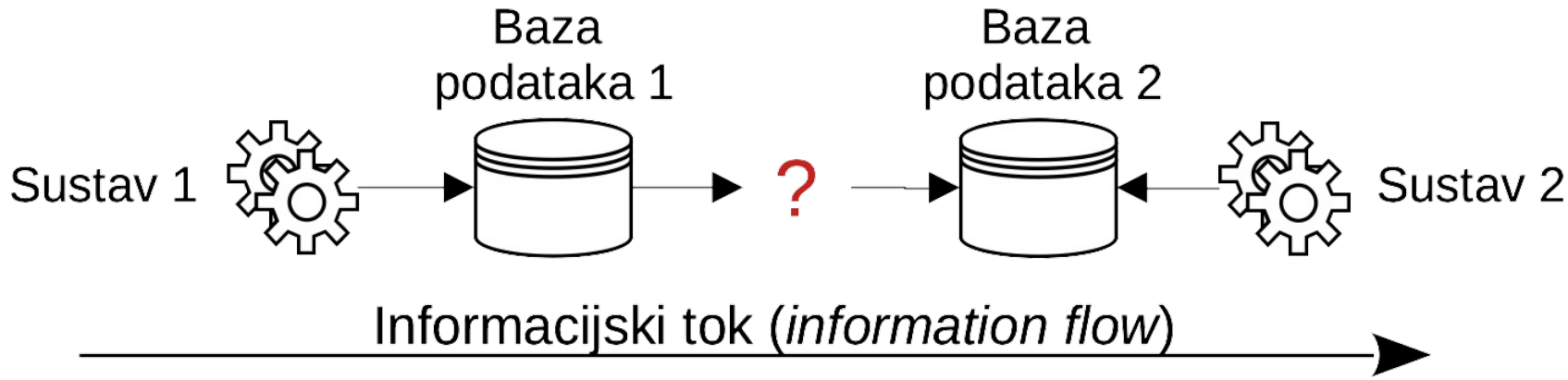


Push i *pull* koncepti u integraciji sustava

- U servisnoj arhitekturi
  - Sustav koji izlaže servis je davatelj (*service provider*)
  - Sustav koji poziva servis je pozivatelj, inicijator, konzumator (*service consumer*)
  - Sam servis određuje
    - Da li inicijator šalje podatke – *push*
    - Da li inicijator dohvaća podatke – *pull*
- Neki sustavi nikad ne iniciraju servisne pozive
  - Zato jer ne znaju korespondentski sustav – nisu dizajnirani na taj način
    - Primjer: baza podataka
  - Svi drugi sustavi oko njih su inicijatori i rade *pull* – dohvaćaju podatke
  - Kako prenijeti neki događaj kroz takav slijed? Prekid lanca prijenosa događaja?

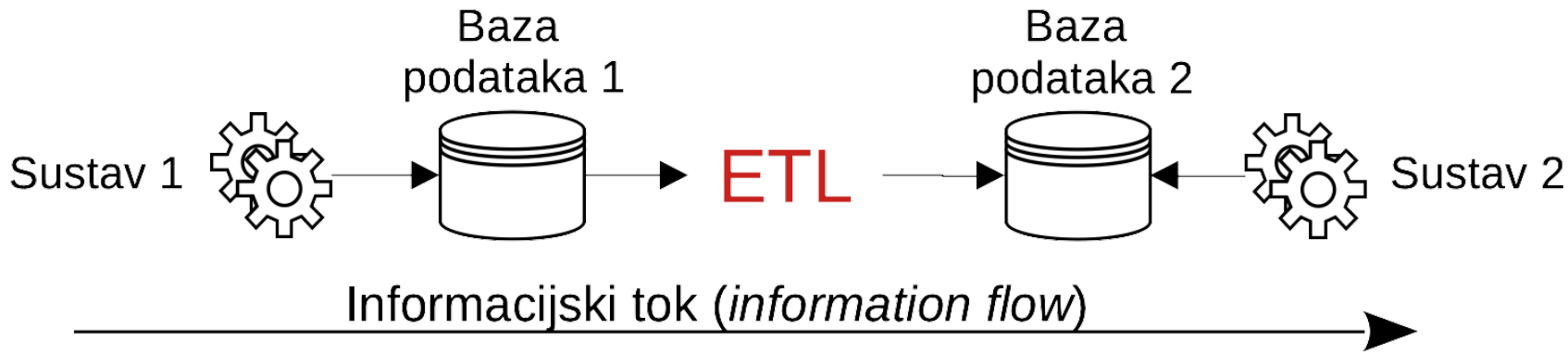


Prekid prijenosa događaja



- Sustav koji se spaja na bazu podataka 1, radi *pull* podataka, zatim se spaja na bazu podataka 2 i radi *push* podataka
- Standardni ETL
  - Extract – Spajanje na bazu 1, dohvat (*pull*)
  - Transform – Transformacija modela (bitno!!!)
  - Load – Spajanje na bazu 2, spremanje (*push*)

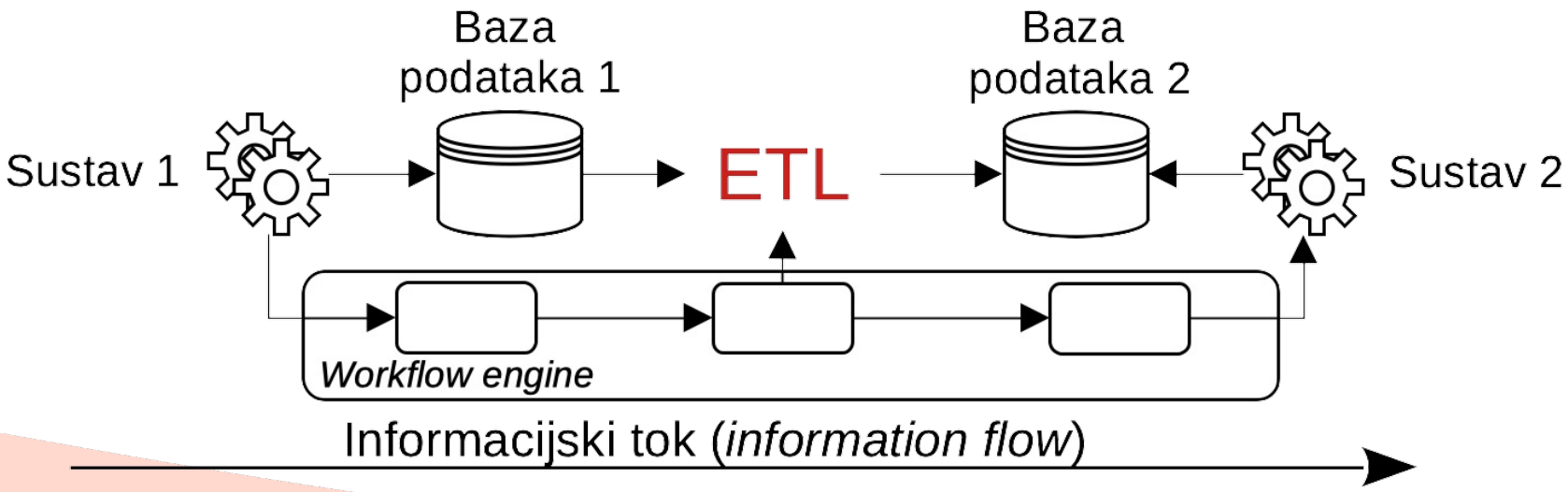
Prekid prijenosa događaja



- No, to nije automatski
  - Insert novog podatka u bazu podataka 1 neće automatski biti prenesen u bazu podataka 2
- Pokretanje ETL-a je asinkrono
  - Scheduling – U pravilnim vremenskim razmacima pokrećemo ETL
  - Polling – Specifično za prijenos događaja – Svako malo provjeravamo da li ima nešto novo na izvoru podataka... ako ima, onda pokrećemo ETL

Strategije uspostave prijenosa događaja

- Paralelna linija prijenosa
  - Nakon upisa podataka u bazu podataka 1, poziva se paralelni proces koji prenosi događaj – *workflow engine*
  - Moguće je da treba i slijed paralelnih procesa za ovu svrhu



Tokovi podataka

- *Publisher-subscriber* model
  - Apache Kafka
  - ne-perzistentno – ako propustite poruku ona se briše
  - uvijek pristupate slijedu podataka – nikad ne čitate točno jedan podatak
  - uglavnom za prijenos podataka
  - *polling* kao glavna metoda prijenosa kod *subscribera*
- Lakše ih je povezati u podatkovni cjevovod
  - Omogućuju filtriranje i upite
  - Čitanje bez brisanja poruke
  - Kod ovakvih sustava treba očekivati slabije performanse nego kod baza podataka – barem što se filtriranja i upita tiče