**Süsteemi üldine kirjeldus**

A diagram of a data flow

Description automatically generated

Joonis 1 Komponent diagramm, sulgudes on Dockeri kontaineri nimetus

Menetlus.API komponent on klientrakendus, mille kasutajaliides on loodud kasutades [SwaggerUI](https://swagger.io/tools/swagger-ui/)-d.

Andmebaasina kasutatakse [PosgreSQL](https://www.postgresql.org/)-i. Menetluse andmed on *menetlus* nimelises andmebaasis. Menetluse toimingute käigus toimuvad ka erinevad ärilised sündmused nagu MenetlusLoodud, MenetlusLoppes, VoetiMenetlusse ja VoetiUlevaatamisele. Need sündmused ja menetluse andmete muutused salvestatakse andmebaasi ühe transaktsiooni sees.

Süsteem implementeerib arendusmustri [Transactional outbox](https://microservices.io/patterns/data/transactional-outbox.html). See on lisakomponent, mida ei olnud proovitöö püstituses, aga on väga oluline andmete terviklikkuse tagamiseks. Menetluse protsess ei sõltu ise RabbitMQ töös olemisest. Lisaks ei saa tekkida olukorda, kus menetluse andmed on muudetud, aga sõnumeid ei suudetud RabbitMQ-sse saata või olukord, kus RabbitMQ-sse on sõnumid saadetud, aga andmebaasi transaktsioon peale seda ebaõnnestub ja sõnumid, mis avaldati, ei toimunud.

[Outboxer](https://github.com/jvalue/outboxer-postgres2rabbitmq) komponent on tarkvara, mis on avalikult olemas github.com koodihoidlas. See komponent kasutab [Debezium](https://debezium.io/) platvormi, mis loeb muudatusi *outbox* tabelist ja publitseerib need RabbitMQ-sse. Docker compose failist saab vaadata, kuidas see komponent seadistatud on.

*Xtee Teavitus Connector* komponent loeb sõnumeid RabbitMQ *xtee-connector* nimelisest *queuest* ja üritab sõnumit saata *Teavitus Teenusele,* mis on *Xtee* komponent, kuhu sõnumid peaksid lõpuks jõudma. Kui sõnumi saatmine ebaõnnestub, proovitakse viis korda uuesti ja iga kord oodatakse kaks korda rohkem kui varasemalt. Alustades kahest sekundist ja lõpetades kolmekümne kahe sekundiga.

Kui ka peale seda sõnumit ei suudeta saata, lisatakse sõnum [Dead Letter Exchangei](https://www.rabbitmq.com/dlx.html) *xtee-connector-dead-letter.* Sealt on võimalik pärast probleemide likvideerimist saata sõnum tagasi esialgsesse *queue*-sse. Rakenduse logidest näeb infot, kui sõnum on saadetud, kui tekib probleeme sõnumi saatmisega ja kui sõnum lõpuks lisati *Dead Letter Exchange’i.*

*Teavitus Teenus* komponent võtab teavitusi vastu *Xtee* protokollile vastavalt, logides maha kättesaadud teavituse. Lisaks on implementeeritud [*Request Deduplication*](https://www.linkedin.com/pulse/request-deduplication-pattern-introduction-ankit-rana/) *arendusmuster.* Kui sõnum sama *teavituse id*-ga peaks jõudma teist korda teenusesse, siis seda ei töödelda ja rakendus logib info, et see sõnum on juba kätte saadud.

Admin komponent on [*Adminier*](https://www.adminer.org/) rakendus, millega saab vaadata PostgreSQL andmebaasi, kasutades sisselogimiseks kasutajat ja parooli admin/admin. Kui arvutis ei ole olemas näiteks *pgAdmin* või muud sarnast tarkvara PostgreSQL andmebaasi haldamiseks.

Selleks, et näha, kuidas süsteem tuleb toime erinevate veaolukordadega, tuleb Dockeris lülitada välja mingi komponent ja jälgida, et peale komponendi käima minemist jõuab sõnum lõppsihtpunkti.

RabbitMQ komponent sisaldab ka kasutajaliidest *Rabbiti*-i rakenduse administreerimiseks, mille saab ligi port 15672. Sisse saab logida kasutades kasutajanime ja parooli guest/guest.

Kui sõnumid on jäänud RabbitMQ *dead letter queue*-sse, siis need saab välja saata avades *dead letter queue* ja sealt valida Destination queue ja vajutades nuppu *Move messages*.

A screenshot of a computer

Description automatically generated