

Modeli komunikacije – Faza 3

Naziv projekta: *Advanced Interactive Painting System (AIPS)*

Autori:

✉ *Veljko Tošić 19438*

✉ *Andrija Stevanović 19382*

Datum: 02.03.2026.

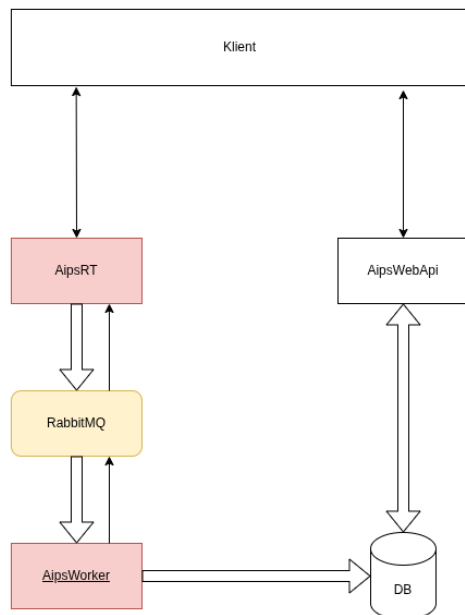
Verzija dokumenta: 0.1

Softverska komponenta

Za komunikaciju između delova sistema opredelili smo se za softversko rešenje **RabbitMQ**.

Na njega se iz .NET backenda konektujemo putem *AMQP* protokola koji je implementiran u *RabbitMQ.Client* nuget package-u.

Delovi sistema koji komuniciraju



Instance koje komuniciraju su AipsRT (Real-time server) i AipsWorker (server za asinhronu obradu).

RT server vrši obradu trenutnog stanja na whiteboardu i komunicira sa klijentima putem web socket-a (SignalR).

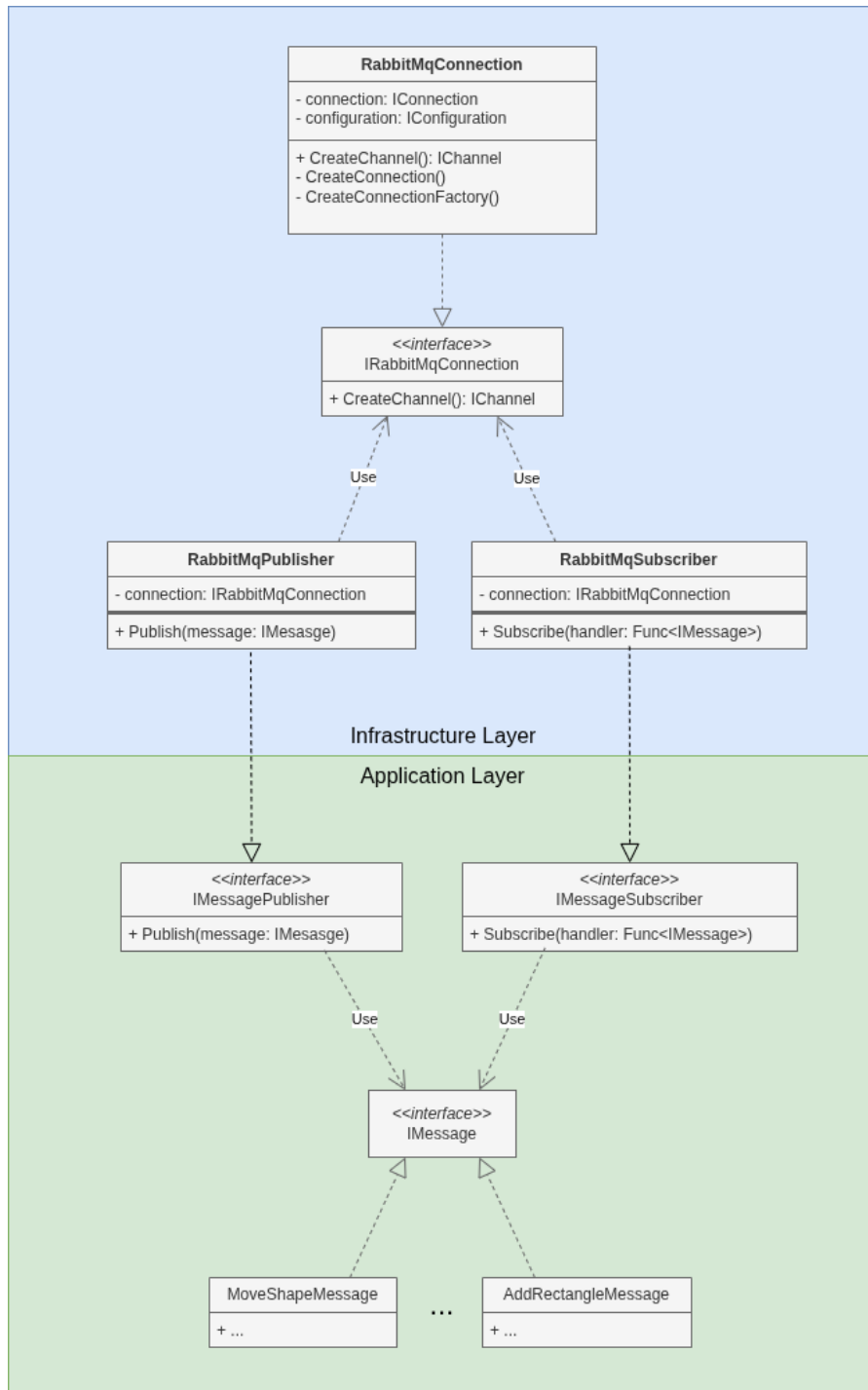
Da bi RT server bio odzivan i imao visoke performanse, on se ne opterećuje komunikacijom sa bazom, već se zahtevi za upis u bazu prosleđuju na MQ.

Sa druge strane worker server prima poruke i izvršava odgovarajuće upise u bazu.

Na taj način Worker i RT serveri mogu horizontalno da se skaliraju nezavisno jedan od drugog, u zavisnosti od potreba.

U slučaju da dođe do greške pri upisu, worker server šalje poruku o grešci na red (*queue*) namenjen za greške, dok RT server prima poruku na tom redu i reaguje na odgovarajući način.

Implementacija



Strana koja šalje poruku koristi *IMessagePublisher*, dok strana koja prima poruku koristi *IMessageSubscriber*.

Za svaki tip poruke (implementira *IMessage*) se kreira red (*queue*).

Deo u aplikacionom layer-u je nezavistan od konkretnog message broker-a koji se koristi.

Na taj način biznis logika ostaje čista od implementaionih detalja. Aplikacioni sloj služi samo da definiše poruke, uz to da može da definiše i način na koji se one obrađuju putem IMessageHandler interfejsa.

U infrastrukturnom layer-u se nalazi implemetacija RabbitMQ konekcije i publisher-a i subscriber-a pomoću iste.