# Communication in Microservices

## Practical Tasks (Technology Agnostic)

There're 3 services: Books, Authors and Frontend (API gateway or backend for frontend pattern).

### Task 1. Request/Response Communication

#### Author Service

Author service possess entities with the following properties: *(Id, FirstName, LastName, Age, Biography, NumberOfBooks)* and has *GET*, *POST* and *PUT* endpoints (*api/v1/authors*) allowing create, change and retrieve all authors.

#### Book Service

Book service possess entities with the following properties: *(Id, AuthorId, Title, Description)* and has *GET* and *POST* endpoints (*api/v1/books*) allowing create and retrieve all books. When new book is being added, *NumberOfBooks* field should be also increased accordingly in the Author service (notice reliability considerations: if increasing number of books in author service fails, book will not be added, and vice-versa).

### Frontend Service

Frontend service should forward calls to authors and books services accordingly. However, for books service it should populate each book (*view-model*) with additional info about author: *(Id, AuthorId, Title, Description, AuthorFirstName, AuthorLastName).*

### Task 2. RPC Communication:

Implement gRPC API for authors and books services. Frontend service exposes only REST API.

### Task 3. Message-Passing Communication:

#### Author Service

Author service sends the following events: *AuthorCreated*, *AuthorChanged*, and also subscribes to *BookCreated* event (when this event fires, authors service should increase *NumberOfBooks* field in corresponding Author entity.

#### Book Service

Book service sends the following events: *BookCreated*, and also subscribes to *AuthorCreated*, *AuthorChanged* events (when these events fire, Book service should add or update basic information about author (*Id, LastName, FirstName*) in its local database.

#### Frontend Service

Frontend service should be simplified: now there is no need to populate books in GET endpoint with additional author info (*FirstName, LastName*), because this info is now stored directly in Book service and populated with events.