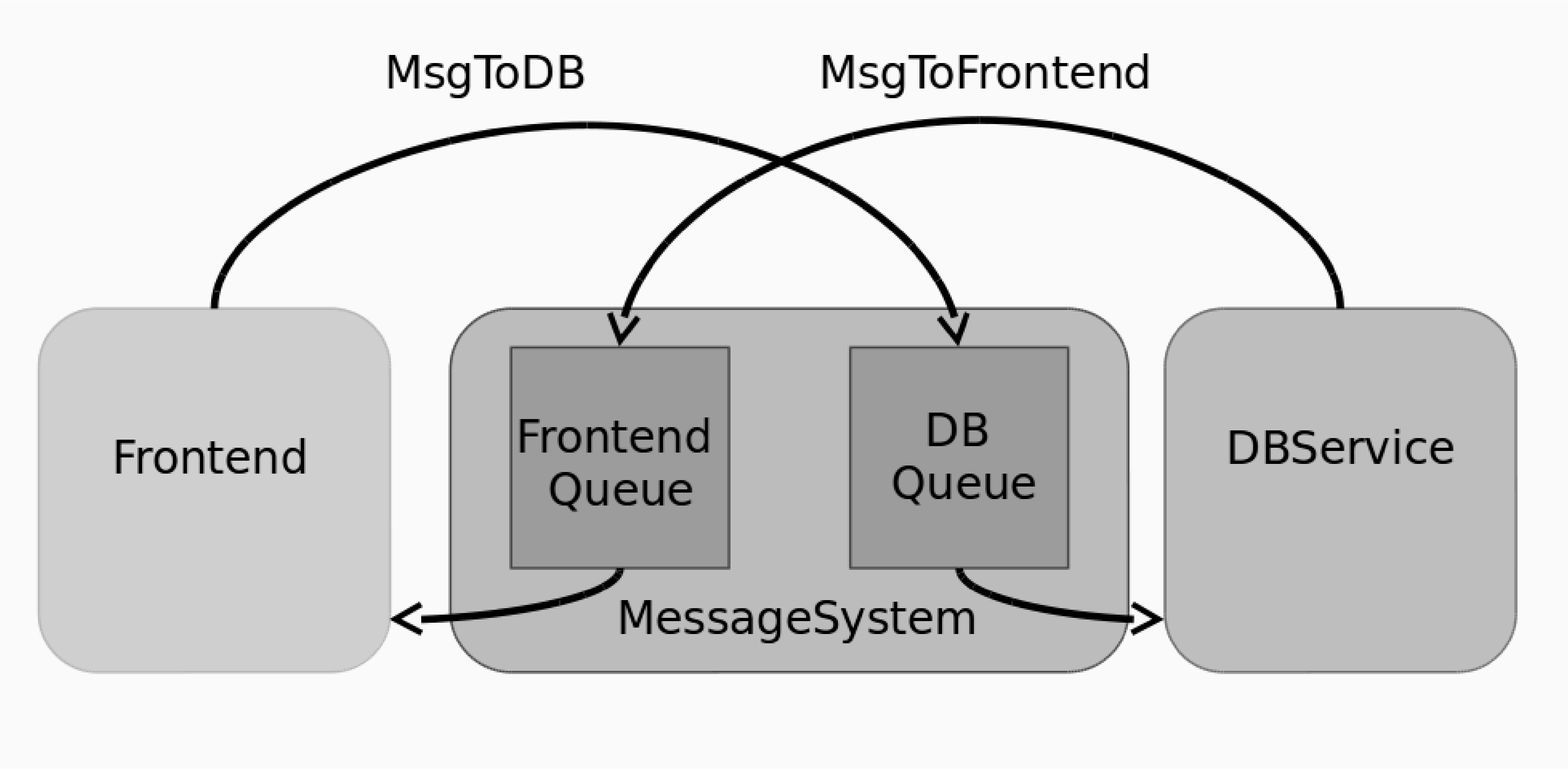
Message System

- Pattern
- Message
- Addressee
- Message system
- MessageSystemContext

Pattern

Message system is like producer-consumer but each Addressee in it can send and receive messages.



Message

Message in an object which is created by one service and processed by another one.

It is effectively thread local.

```
public abstract class Message {
    private final Address from;
    private final Address to;
    public Message(Address from, Address to) {
        this.from = from;
        this.to = to;
    }
    public Address getFrom() {
        return from;
    }
    public Address getTo() {
        return to;
    }
    public abstract void exec(Addressee addressee);
}
```

Addressee

Addressee is an object which can receive messages.

```
public class DBServiceImpl implements DBService, Addressee {
    private final Address address;
    private final MessageSystemContext context;

    public DBServiceImpl(MessageSystemContext context, Address address) {
        this.context = context;
        this.address = address;
    }

    public void init() {
        context.getMessageSystem().addAddressee(this);
    }

    @Override
    public Address getAddress() {
        return address;
    }

    public int getUserId(String name) {
        //todo: load id from db
        return name.hashCode();
    }
}
```

MessageSystem

Message system is an object which contains a queue for messages for each Addressee.

Can add messages to a queue.

Execute scheduled messages in a loop

```
public void sendMessage(Message message) {
    messagesMap.get(message.getTo()).add(message);
}

public void start() {
    for (Map.Entry<Address, Addressee> entry : addresseeMap.entrySet()) {
        new Thread(() -> {
            while (true) {
                ConcurrentLinkedQueue<Message> queue = messagesMap.get(entry.getKey());
                while (!queue.isEmpty()) {
                    Message message = queue.poll();
                    message.exec(entry.getValue());
                }
                try {
                    Thread.sleep(MessageSystem.DEFAULT_STEP_TIME);
                } catch (InterruptedException e) {
                    e.printStackTrace();
                }
            }
        }).start();
    }
}
```

MessageSystemContext

Context with

- message system
- addresses of all services

MessageSystemContext is application specific.