Tobias Meurer Sven Dettmers Milena Dreier

BAIP Aufgabe 4

HAPSAR Adapter

Transportdienstleister Adapter

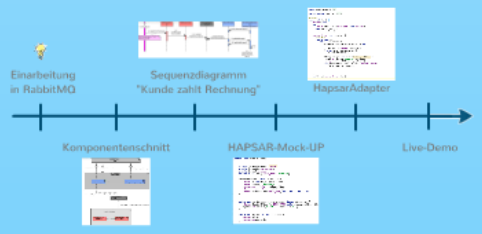
Tobias Meurer Sven Dettmers Milena Dreier

BAIP Aufgabe 4

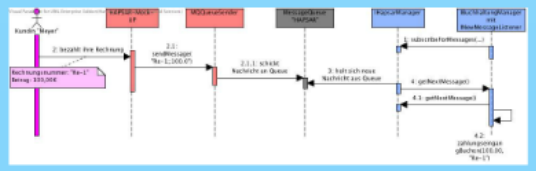
HAPSAR Adapter

Transportdienstleister Adapter

HAPSAR Adapter



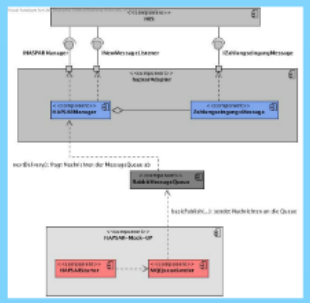
Einarbeitung
in RabbitMQ



Sequenzdiagramm
"Kunde zahlt Rechnung"

HapsarAdapter

Komponentenschnitt

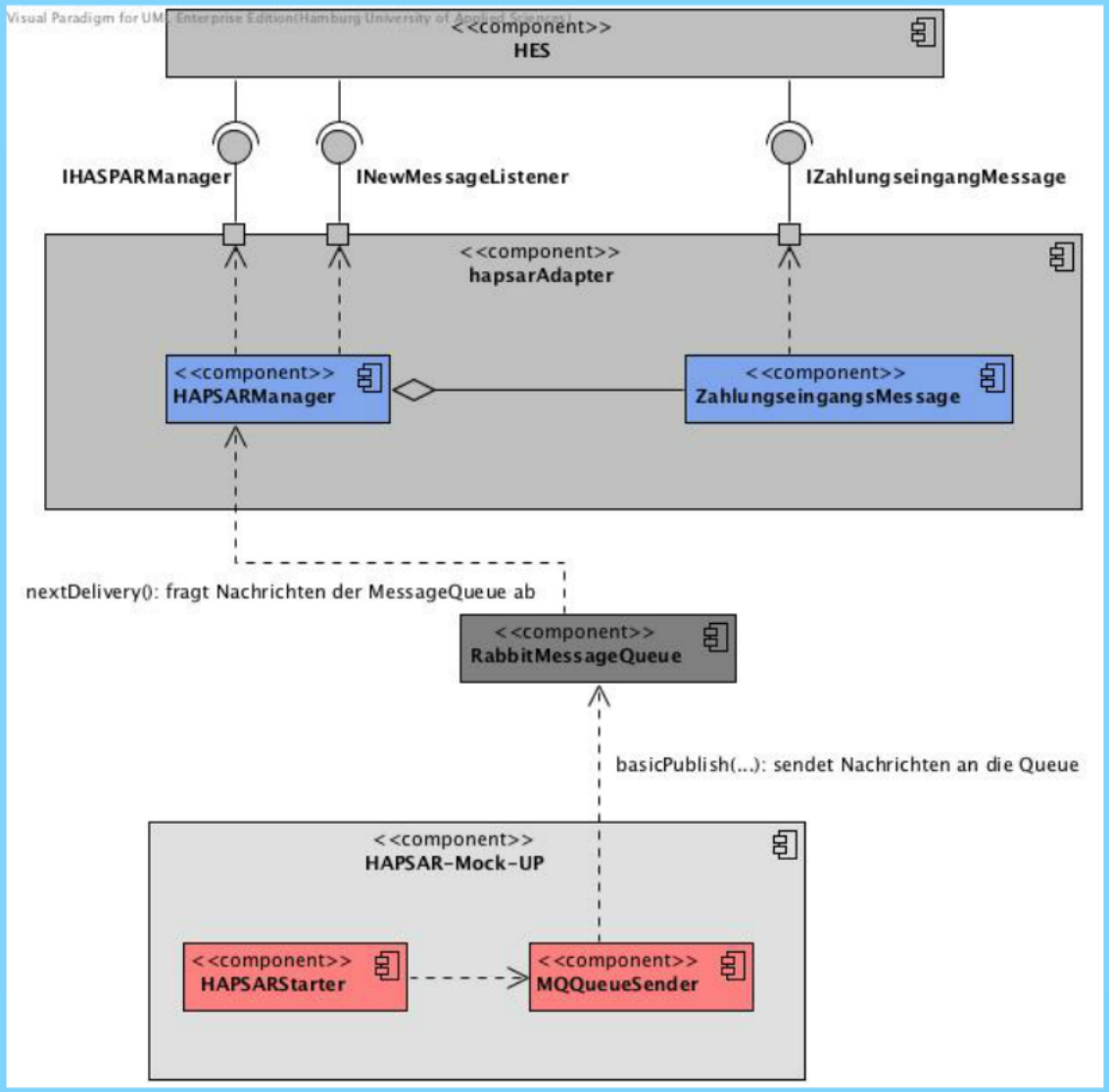


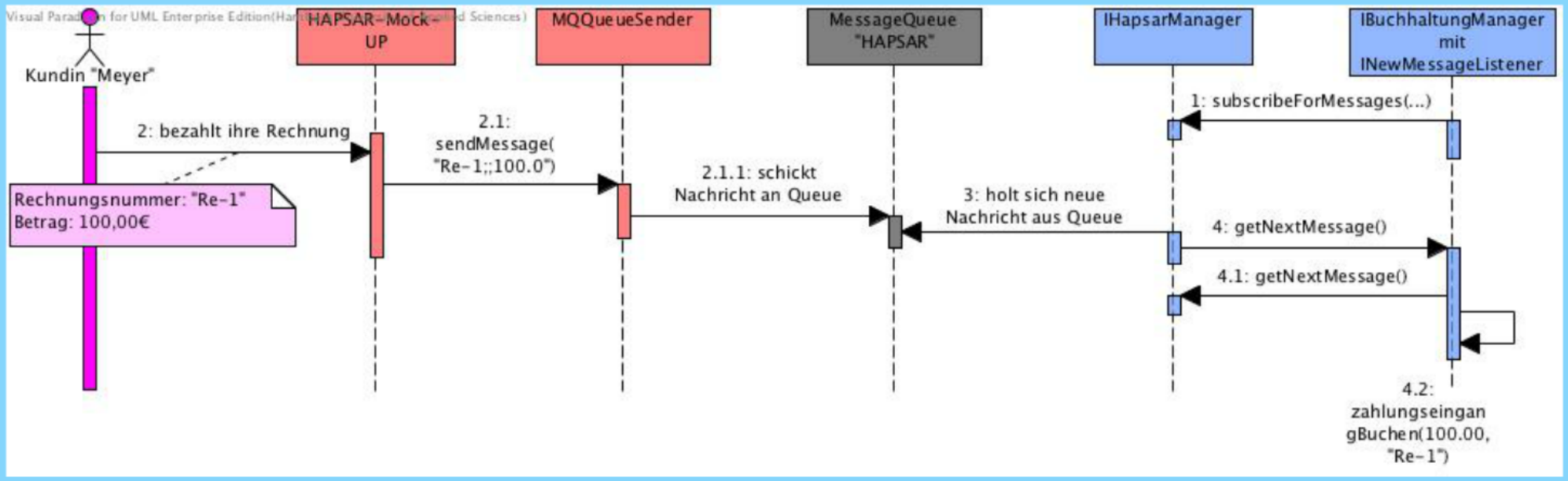
HAPSAR-Mock-UP

```
public class MQQueueSender{

    private ConnectionFactory factory;
    private Connection connection;
    private Channel channel;
    private final String QUEUE_NAME = "HAPSAR";

    MQQueueSender() throws IOException {
        factory = new ConnectionFactory();
        factory.setHost("localhost");
        connection = factory.newConnection();
        channel = connection.createChannel();
        channel.queueDeclare(QUEUE_NAME, true, false, false, null);
    }

    public void sendMessage(String message) throws IOException {
        channel.basicPublish("", QUEUE_NAME, null, message.getBytes());
        System.out.println(" Sent '" + message + "'");
    }

    public void close() throws IOException {
        channel.close();
        connection.close();
    }
```

Live-Demo

Komponentenschnitt





Sequenzdiagramm "Kunde zahlt Rechnung"

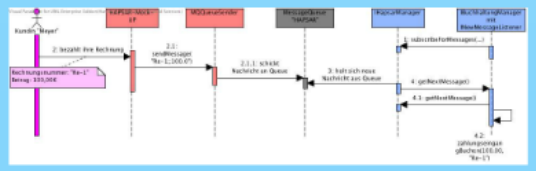
HAPSAR-Mock-UP

```java
public class MQQueueSender{


    private ConnectionFactory factory;
    private Connection connection;
    private Channel channel;
    private final String QUEUE_NAME = "HAPSAR";

    MQQueueSender() throws IOException {
        factory = new ConnectionFactory();
        factory.setHost("localhost");
        connection = factory.newConnection();
        channel = connection.createChannel();
        channel.queueDeclare(QUEUE_NAME, true, false, false, null);
    }


    public void sendMessage(String message) throws IOException {
        channel.basicPublish("", QUEUE_NAME, null, message.getBytes());
        System.out.println(" Sent '" + message + "'");
    }

    public void close() throws IOException {
        channel.close();
        connection.close();
    }
```

```java
public HAPSARManager() {...}

private boolean isReceive() {...}

@Override
public void start() {
    receive = true;
    try {
        channel.basicConsume(QUEUE_NAME, true, consumer);
    } catch (IOException e) {
        e.printStackTrace();
    }
    receiverThread = new Runnable() {
        @Override
        public void run() {
            while (isReceive()) {
                QueueingConsumer.Delivery delivery = null;
                try {
                    delivery = consumer.nextDelivery();
                } catch (InterruptedException e) {
                    e.printStackTrace();
                }
                String message = new String(delivery.getBody());
                System.out.println("Received: "+message);
                messages.add(new ZahlungseingangMessage(message));
                for (INewMessageListener listener : messageListener){
                    listener.getNextMessage();
                }
            }
        }
    };
    new Thread(receiverThread).start();
}
```

HapsarAdapter

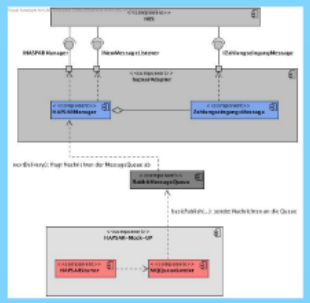
Einarbeitung
in RabbitMQ



Sequenzdiagramm
"Kunde zahlt Rechnung"

HapsarAdapter

Komponentenschnitt



HAPSAR-Mock-UP

```
public class MQQueueSender{

    private ConnectionFactory factory;
    private Connection connection;
    private Channel channel;
    private final String QUEUE_NAME = "HAPSAR";

    MQQueueSender() throws IOException {
        factory = new ConnectionFactory();
        factory.setHost("localhost");
        connection = factory.newConnection();
        channel = connection.createChannel();
        channel.queueDeclare(QUEUE_NAME, true, false, false, null);
    }

    public void sendMessage(String message) throws IOException {
        channel.basicPublish("", QUEUE_NAME, null, message.getBytes());
        System.out.println(" Sent '" + message + "'");
    }

    public void close() throws IOException {
        channel.close();
        connection.close();
    }
```

Live-Demo

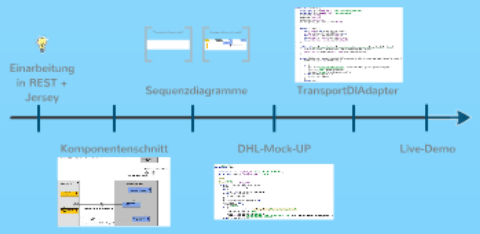
Tobias Meurer Sven Dettmers Milena Dreier

BAIP Aufgabe 4

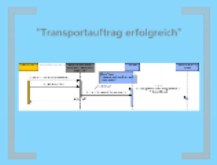
HAPSAR Adapter

Transportdienstleister Adapter

Transportdienstleister Adapter



Einarbeitung in REST + Jersey

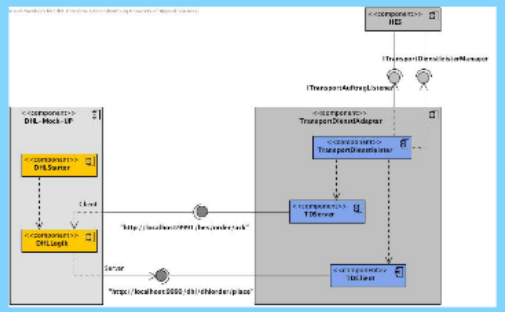


Sequenzdiagramme

```
public class TDClient {
    static final String REST_URI = "http://localhost:9998/dhl/";
    static final String PLACE_PATH = "dhlorder/place";
    ClientConfig config;
    WebResource service;
    WebResource subService;
    com.sun.jersey.api.client.Client client;


    public TDClient() {
        this.config = new DefaultClientConfig();
        config.getClasses().add(JSONObject.class);
        this.client = Client.create(config);
        this.service = client.resource(REST_URI);
        this.subService = service.path(PLACE_PATH);
    }

    public void sendeTransportauftrag(TransportauftragTyp transportauftrag, KundenTyp kunde) {
        JSONObject json = transportTypen2uJSON(transportauftrag, kunde);
        subService.type(MediaType.TEXT_PLAIN).post(json.toString());
    }

    private JSONObject transportTypen2uJSON(TransportauftragTyp transportauftrag, KundenTyp kunde) {
        JSONObject json = new JSONObject();

        try {
            json.put("transportauftragsNummer", transportauftrag.getTransportauftragNr());
            json.put("name", kunde.getName());
            json.put("adresse", kunde.getAdresse());
        } catch (JSONException e) {
            e.printStackTrace();
        }
        return json;
```

TransportDlAdapter

Komponentenschnitt

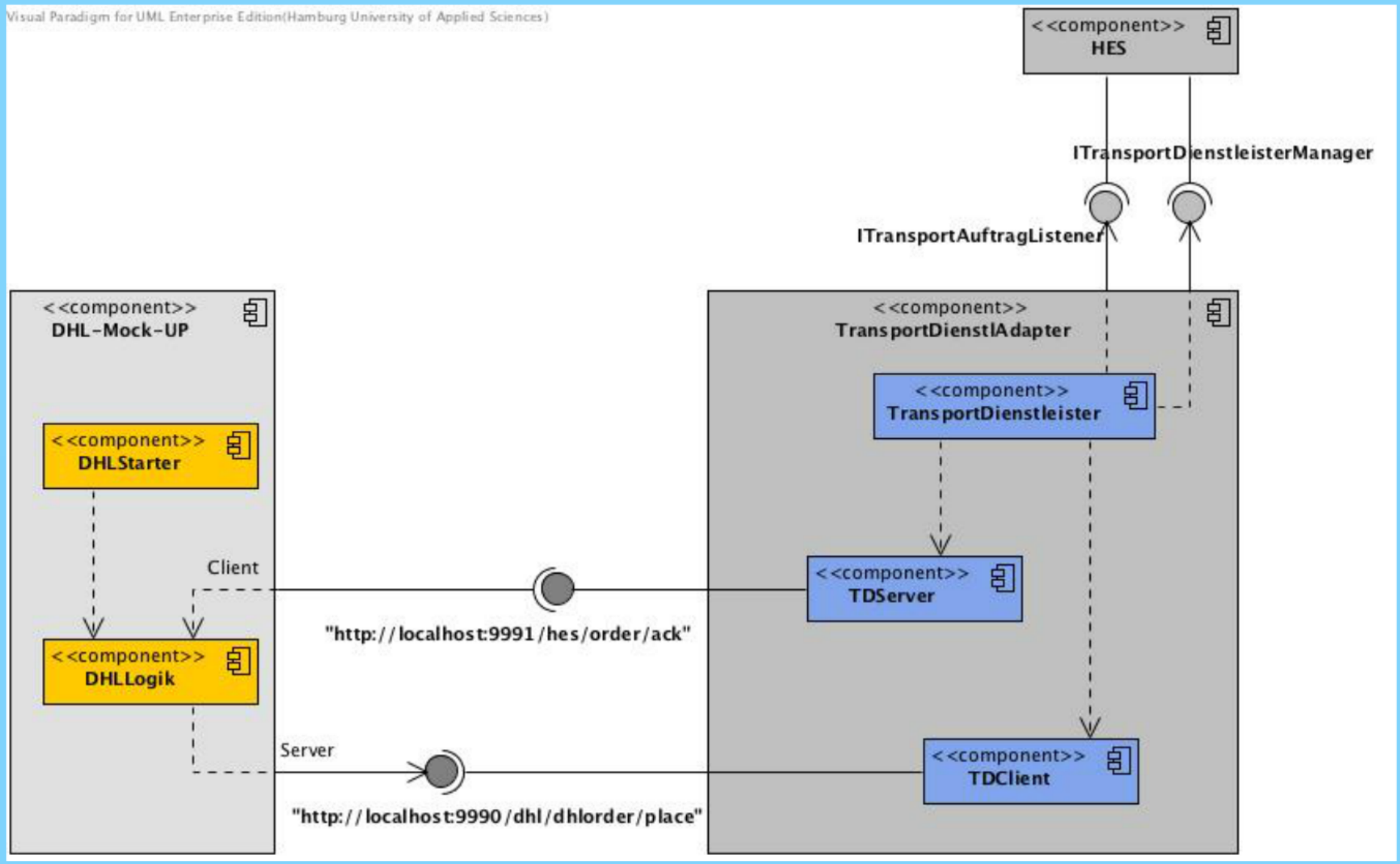


DHL-Mock-UP

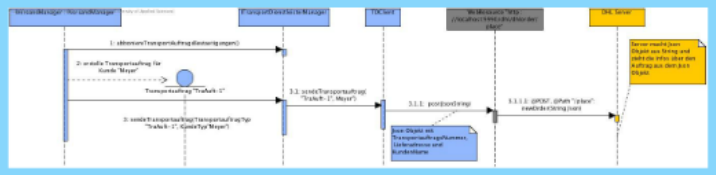
```
@Path("/dhlorder")
public class Logic {

    static final String AMS_REST_URI = "http://localhost:9091/hes";
    static final String AMS_ORDERACK_PATH = "order/ack";

    @POST
    @Path("/place")
    @Consumes(MediaType.TEXT_PLAIN)
    public void newOrder(String orderString) {
        JSONObject order = null;
        try {
            order = new JSONObject(orderString);
        } catch (JSONException e) {
            e.printStackTrace();
        }
        String transportauftragsNummer = null;
        String name = null;
        String adresse = null;
        try {
            transportauftragsNummer = order.getString("transportauftragsNummer");
            name = order.getString("name");
            adresse = order.getString("adresse");
            System.out.printf("Transportauftrag erhalten für: %s %s %s\n", name, adresse, transportauftragsNummer);
        } catch (JSONException e) {
```

Live-Demo

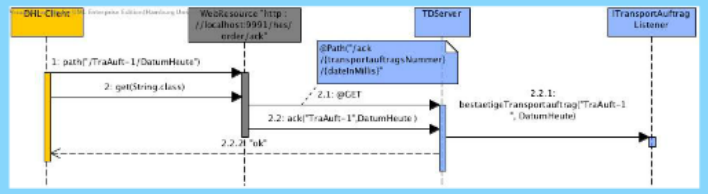
Komponentenschnitt



"Transportauftrag senden"

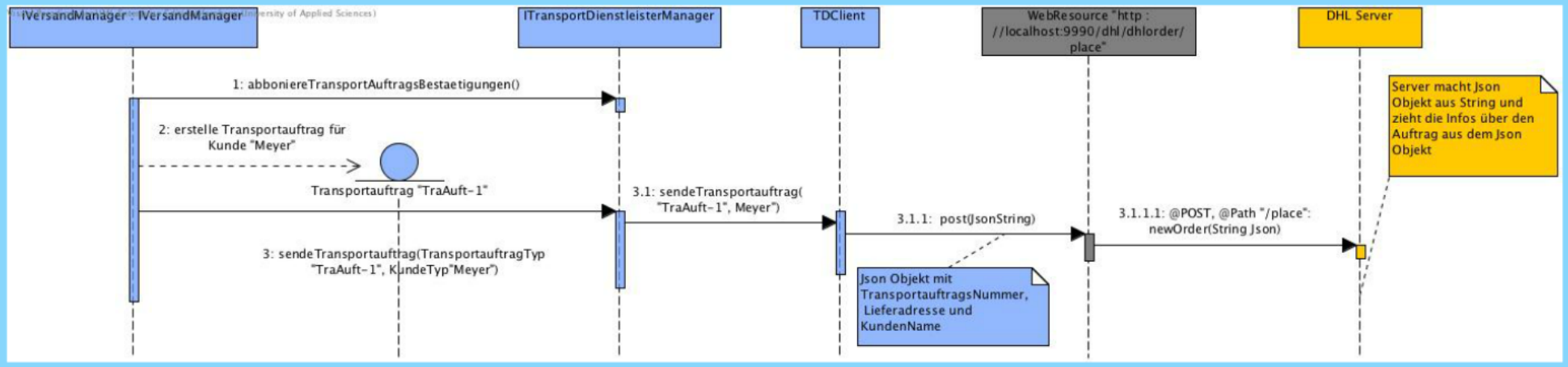


"Transportauftrag erfolgreich"

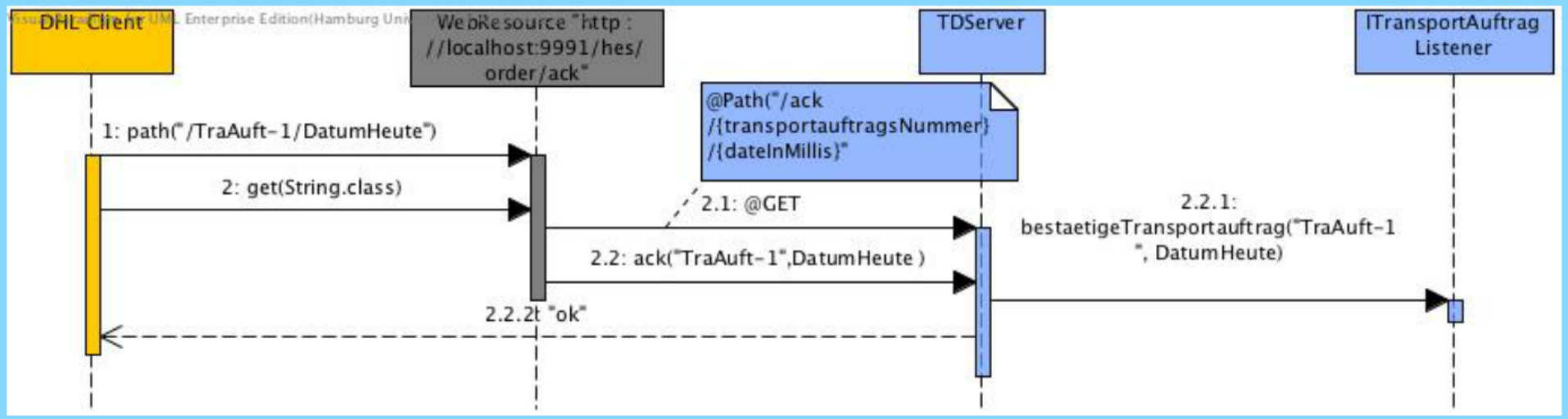


Sequenzdiagramme

"Transportauftrag senden"



"Transportauftrag erfolgreich"



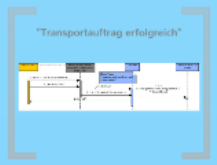
DHL-Mock-UP

```java
@Path("/dhlorder")
public class Logic {

    static final String HES_REST_URI = "http://localhost:9991/hes";
    static final String HES_ORDERACK_PATH = "order/ack";

    @POST
    @Path("/place")
    @Consumes(MediaType.TEXT_PLAIN)
    public void newOrder(String orderString) {
        JSONObject order = null;
        try {
            order = new JSONObject(orderString);
        } catch (JSONException e) {
            e.printStackTrace();
        }
        String transportauftragsNummer = null;
        String name = null;
        String adresse = null;
        try {
            transportauftragsNummer = order.getString("transportauftragsNummer");
            name = order.getString("name");
            adresse = order.getString("adresse");
            System.out.printf("Transportauftrag erhalten für: %s %s %s%n", name, adresse, transportauftragsNummer);
        } catch (JSONException e) {
```

```java
public class TDClient {
    static final String REST_URI = "http://localhost:9990/dhl/";
    static final String PLACE_PATH = "dhlorder/place";
    ClientConfig config;
    WebResource service;
    WebResource subService;
    com.sun.jersey.api.client.Client client;


    public TDClient() {
        this.config = new DefaultClientConfig();
        config.getClasses().add(JSONObject.class);
        this.client = Client.create(config);
        this.service = client.resource(REST_URI);
        this.subService = service.path(PLACE_PATH);
    }

    public void sendeTransportauftrag(TransportauftragTyp transportauftrag, KundenTyp kunde) {
        JSONObject json = transportTypenZuJSON(transportauftrag, kunde);
        subService.type(MediaType.TEXT_PLAIN).post(json.toString());
    }

    private JSONObject transportTypenZuJSON(TransportauftragTyp transportauftrag, KundenTyp kunde) {
        JSONObject json = new JSONObject();

        try {
            json.put("transportauftragsNummer", transportauftrag.getTransportauftragNr());
            json.put("name", kunde.getName());
            json.put("adresse", kunde.getAdresse());
        } catch (JSONException e) {
            e.printStackTrace();
        }
        return json;
```

TransportDIAdapter

Einarbeitung in REST + Jersey

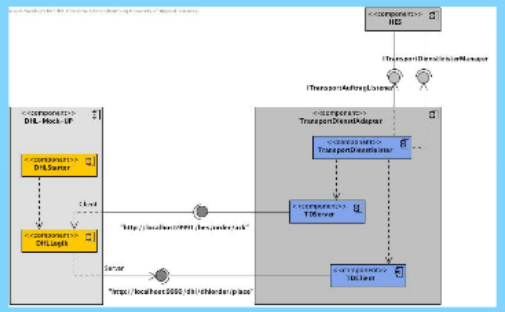


Sequenzdiagramme

```
public class TDClient {
    static final String REST_URI = "http://localhost:9998/dhl/";
    static final String PLACE_PATH = "dhlorder/place";
    ClientConfig config;
    WebResource service;
    WebResource subService;
    com.sun.jersey.api.client.Client client;


    public TDClient() {
        this.config = new DefaultClientConfig();
        config.getClasses().add(JSONObject.class);
        this.client = Client.create(config);
        this.service = client.resource(REST_URI);
        this.subService = service.path(PLACE_PATH);
    }

    public void sendeTransportauftrag(TransportauftragTyp transportauftrag, KundenTyp kunde) {
        JSONObject json = transportTypen2uJSON(transportauftrag, kunde);
        subService.type(MediaType.TEXT_PLAIN).post(json.toString());
    }

    private JSONObject transportTypen2uJSON(TransportauftragTyp transportauftrag, KundenTyp kunde) {
        JSONObject json = new JSONObject();

        try {
            json.put("transportauftragsNummer", transportauftrag.getTransportauftragNr());
            json.put("name", kunde.getName());
            json.put("adresse", kunde.getAdresse());
        } catch (JSONException e) {
            e.printStackTrace();
        }
        return json;
```

TransportDlAdapter

Komponentenschnitt



DHL-Mock-UP

```
@Path("/dhlorder")
public class Logic {

    static final String AMS_REST_URI = "http://localhost:9091/hes";
    static final String AMS_ORDERACK_PATH = "order/ack";

    @POST
    @Path("/place")
    @Consumes(MediaType.TEXT_PLAIN)
    public void newOrder(String orderString) {
        JSONObject order = null;
        try {
            order = new JSONObject(orderString);
        } catch (JSONException e) {
            e.printStackTrace();
        }
        String transportauftragsNummer = null;
        String name = null;
        String adresse = null;
        try {
            transportauftragsNummer = order.getString("transportauftragsNummer");
            name = order.getString("name");
            adresse = order.getString("adresse");
            System.out.printf("Transportauftrag erhalten für: %s %s %s\n", name, adresse, transportauftragsNummer);
        } catch (JSONException e) {
```

Live-Demo