



Public Transportation: S-Trains

02268 Final Presentation • Group 15



Domain description

211 million trips in 2021

High reliance

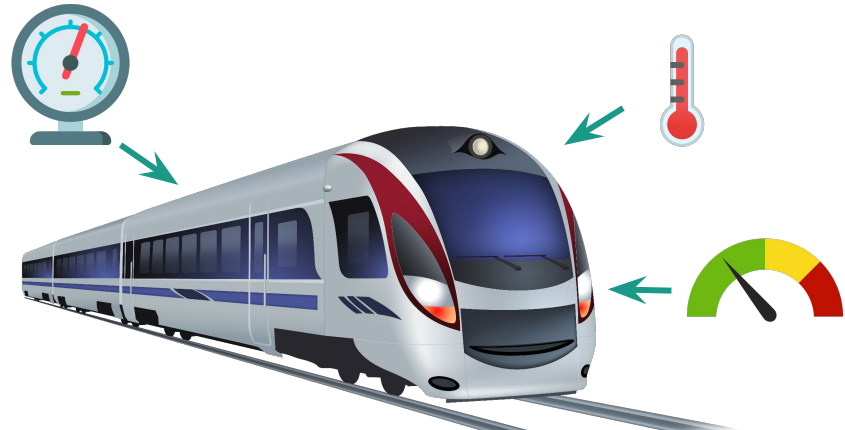
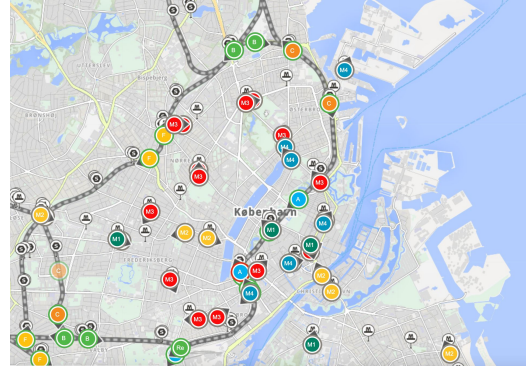
Complex systems



Events

Event Sources

- Rejseplan API
- Train Sensor Simulator



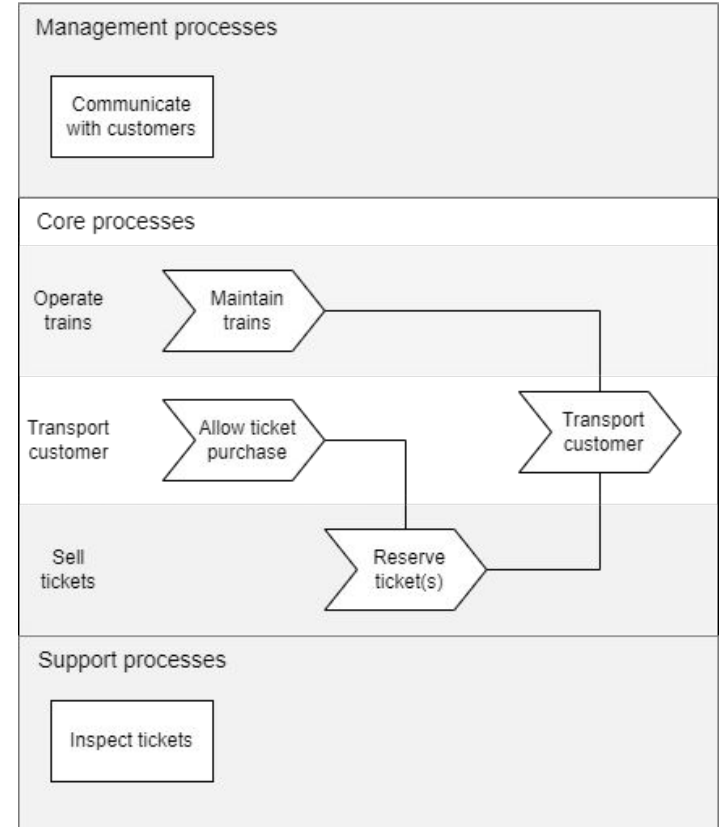
Event Processing



Processes

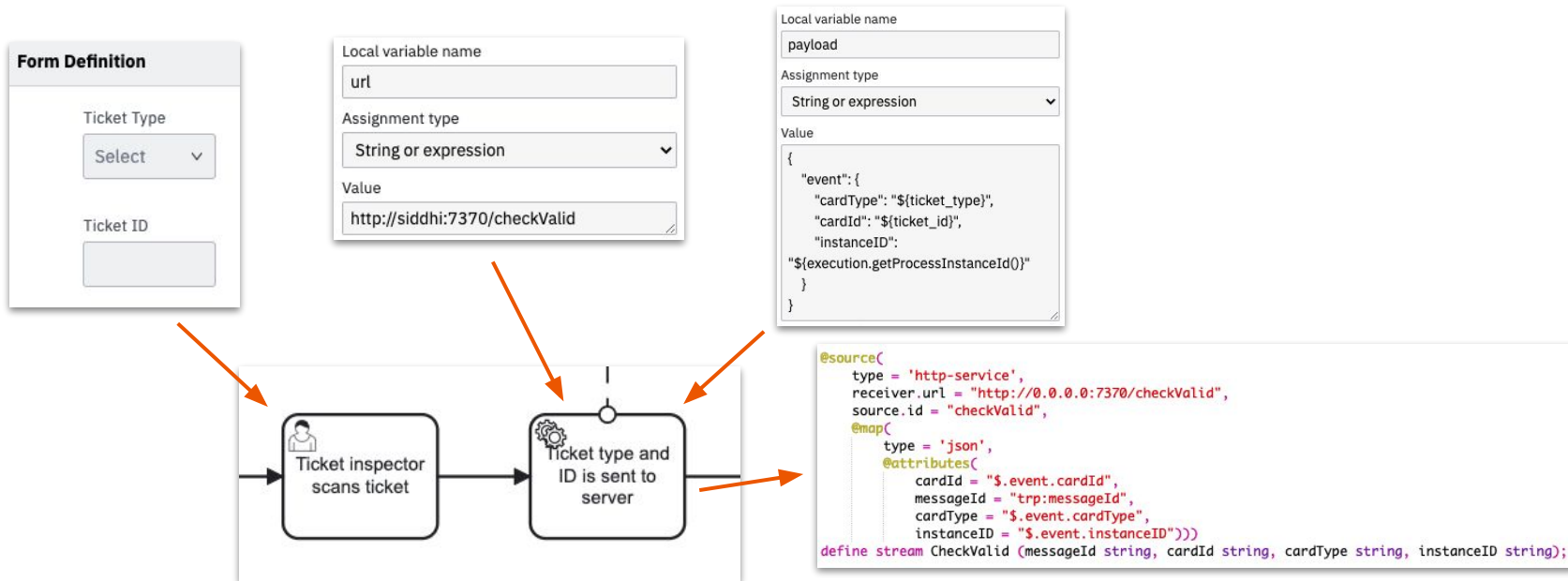
Process landscape

- Standard Customer Journey
- Typical S-train Journey
- Reserve Tickets
- Inspect Tickets
- Alert for Train Maintenance
- Manage Train Delays
- Allow Ticket Purchase



Interaction between processes and events

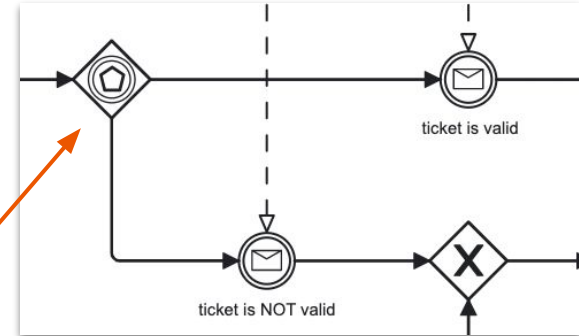
Camunda to Siddhi



Siddhi to Camunda

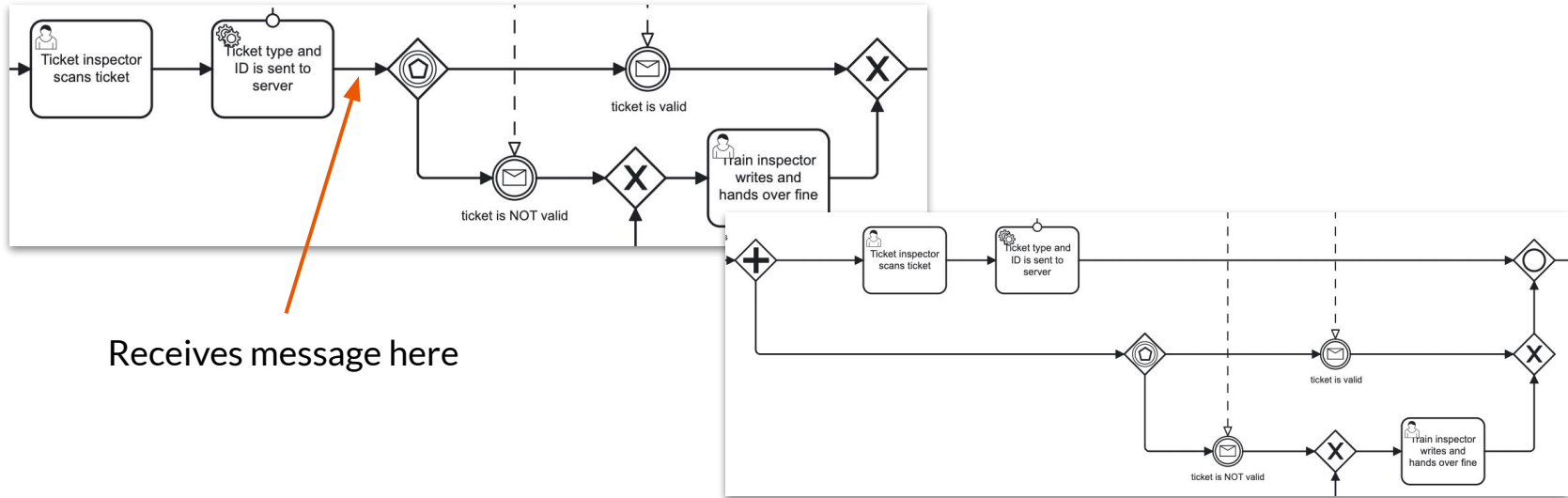
```
@info(name = 'CheckTicketStream')
from CheckValid[cardType == 'ticket']
left outer join ticketsDB
on CheckValid.cardId == ticketsDB.id
select
  CheckValid.messageId,
  ifThenElse(ticketsDB.id is null, "invalid_ticket", "valid_ticket") as messageName,
  CheckValid.instanceID as instanceID
insert into CheckValidResponseStream;
```

```
@sink(
  type = 'http',
  publisher.url = 'http://camunda:8080/engine-rest/message',
  headers = "'Host:camunda:8080'",
  method = 'POST',
  @map(
    type = 'json',
    @payload("""{"messageName": "{{messageName}}", "processInstanceId": "{{instanceID}}"}"""))
define stream CheckValidResponseStream (messageId string, messageName string, instanceID string);
```



Challenges, take-home messages, and conclusions

Challenge: Camunda Race Conditions



Receives message here

Conclusions

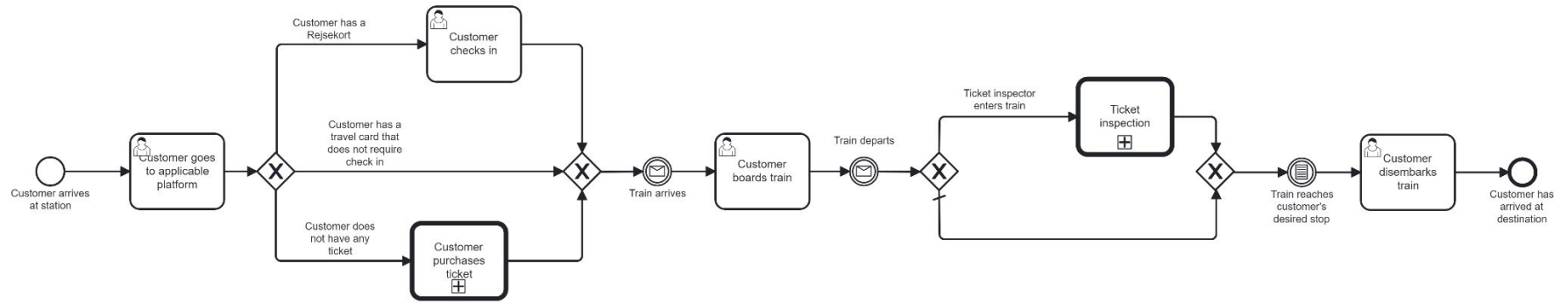
Identifying business processes

Event Sources

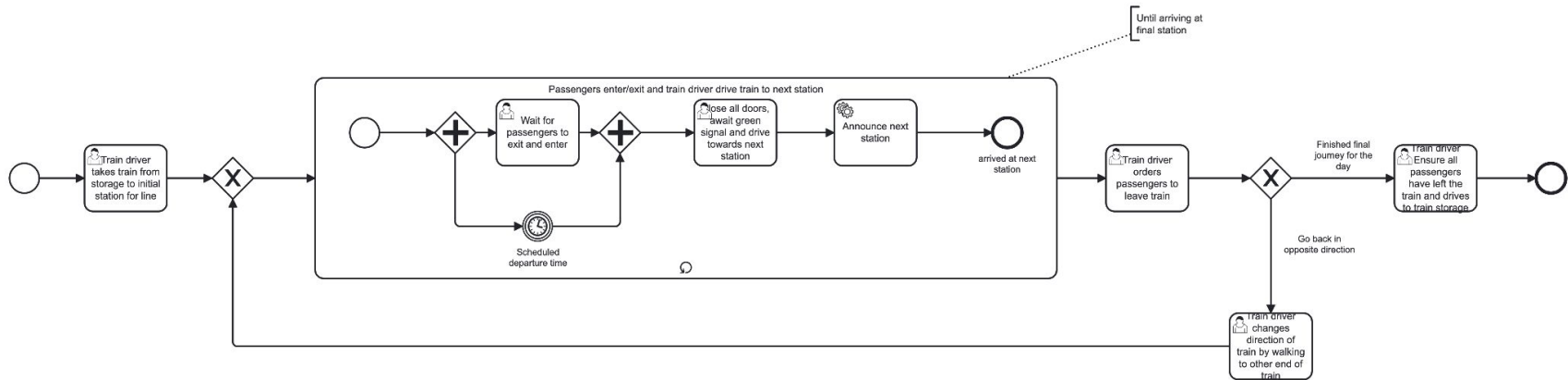
Event Processes



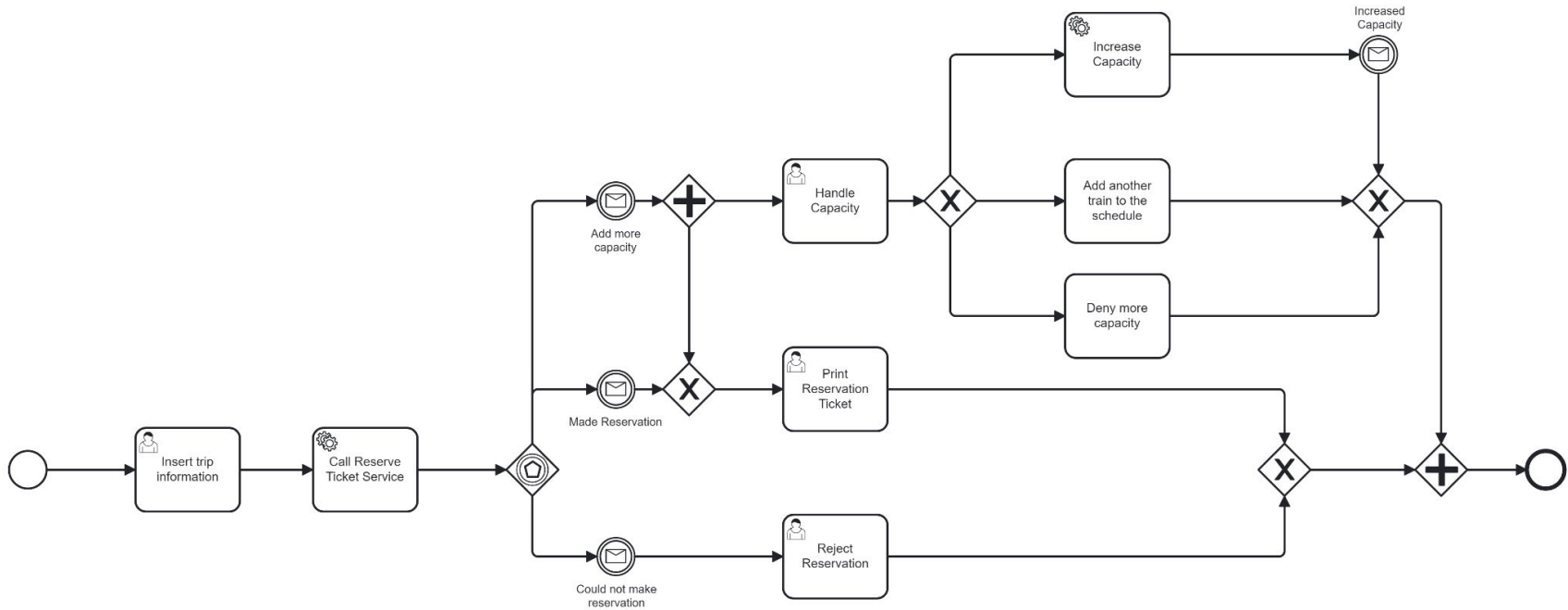
Appendix



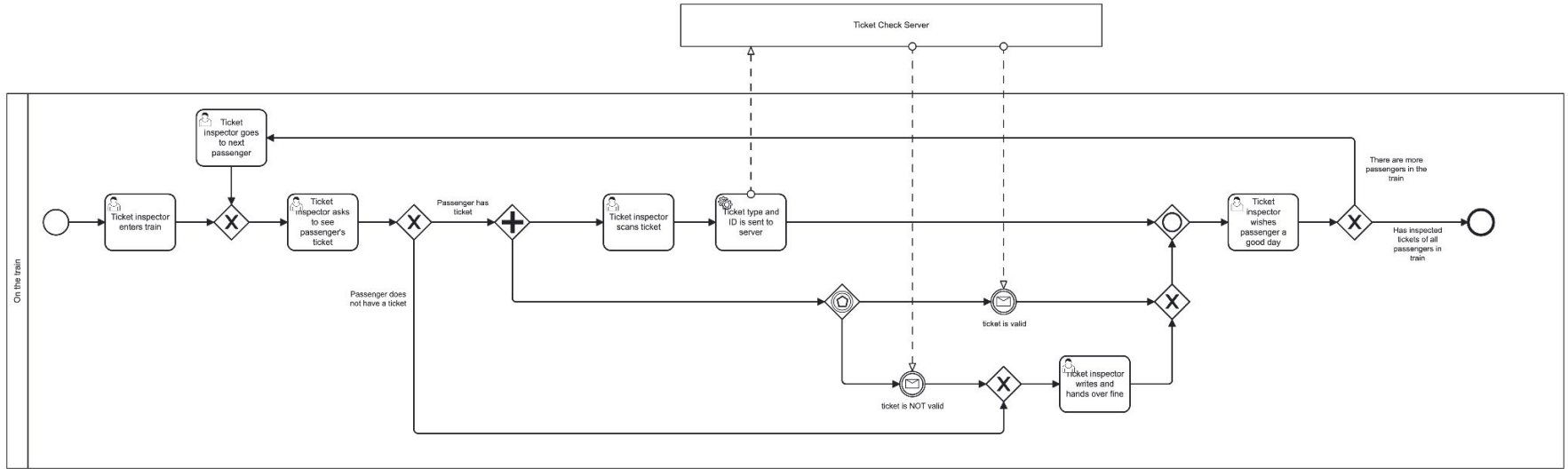
Standard Customer Journey



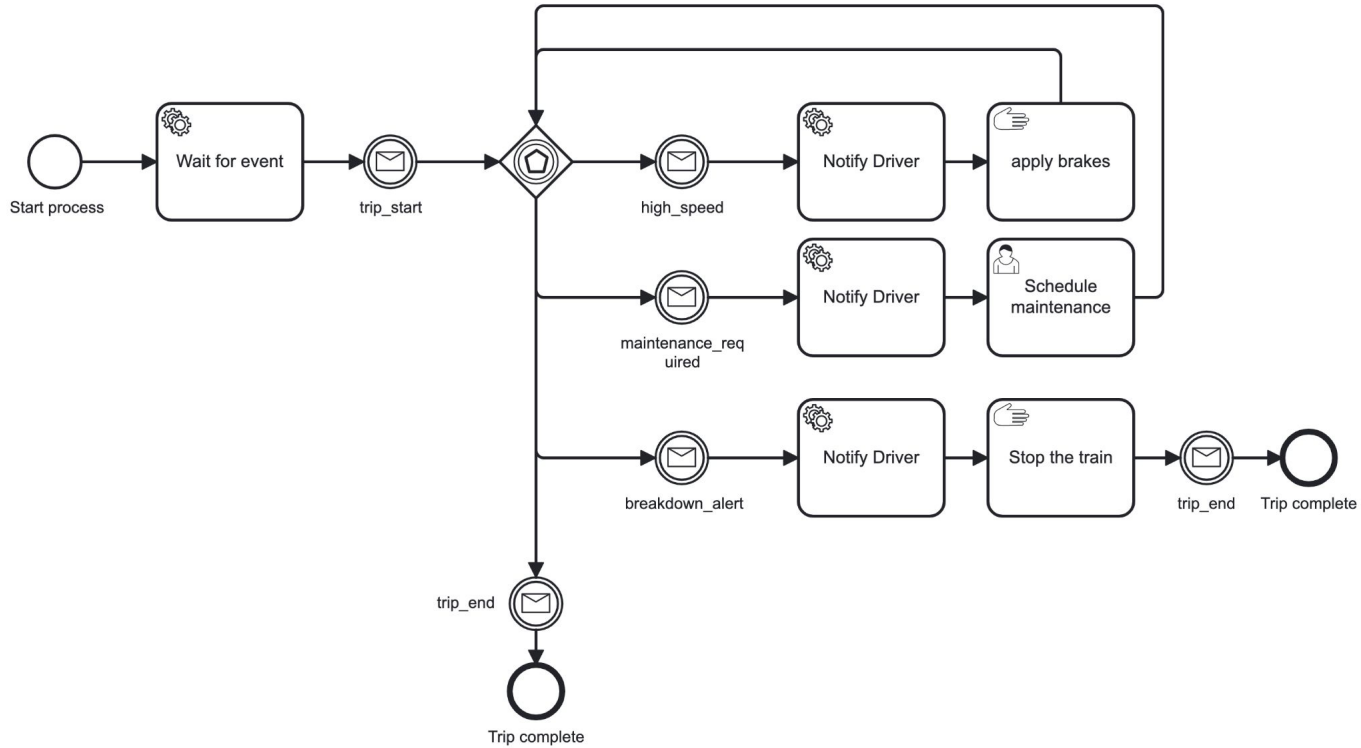
Typical S-train journey



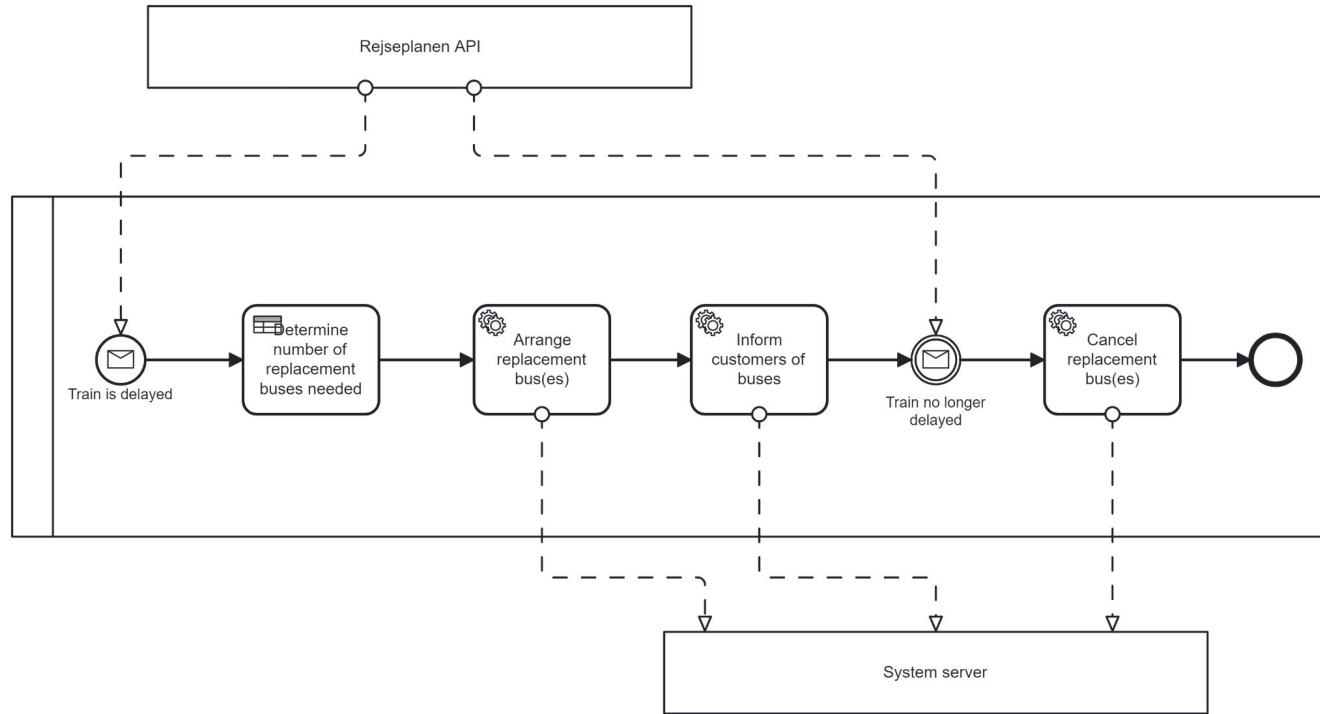
Reserve Tickets



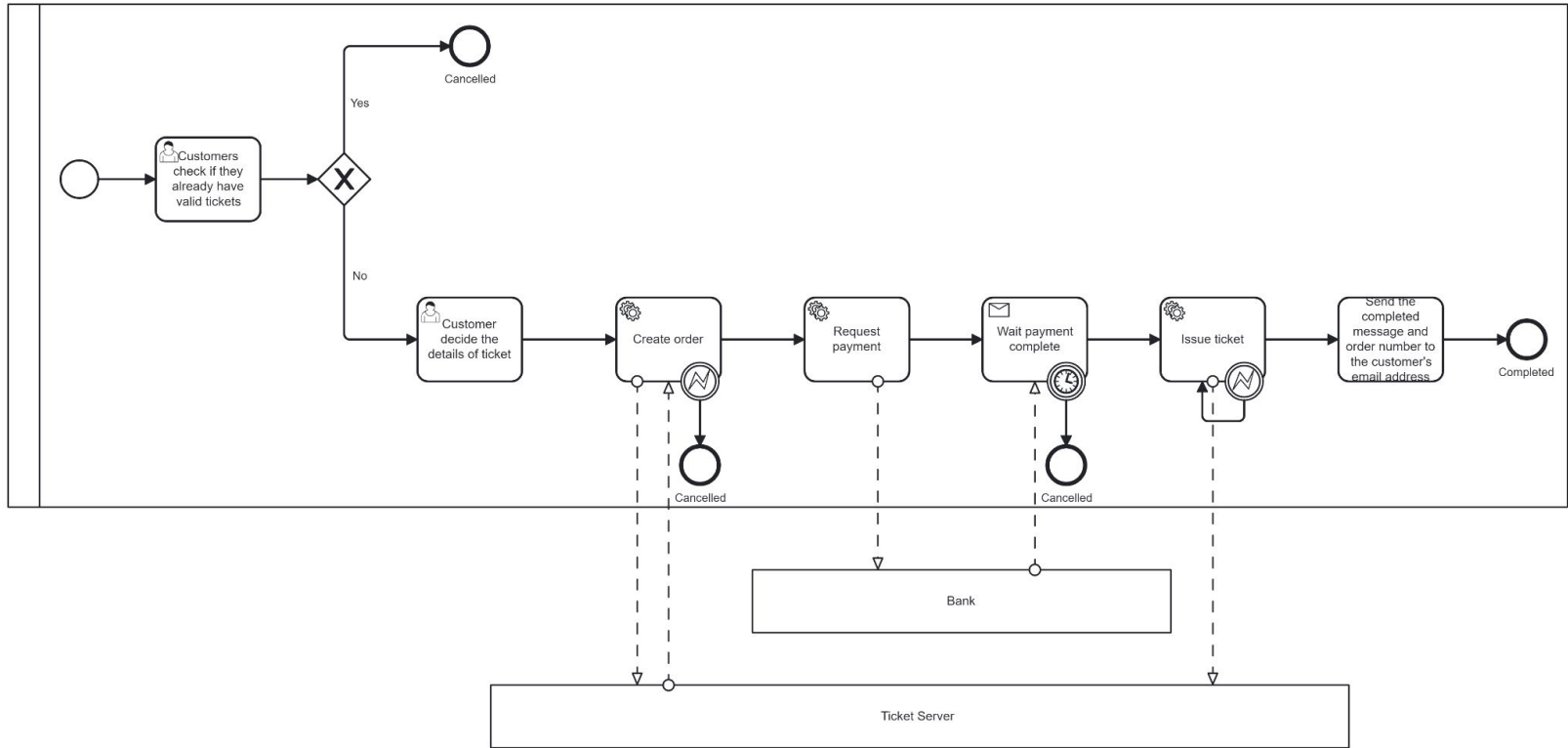
Inspect tickets



Alert Train Maintenance



Manage Train Delays



Allow ticket purchase