Safe Railway-crossing for autonomous vehicles

A chaincode to manage the crossing of autonomous vehicles and trains at unguarded level crossings.

Chaincode Endpoints

- crossingExists(crossingId):boolean
  returns true if a crossing exitst with the specified Id
- readCrossing(crossingId):Crossing
  if it exits, it returns the crossing with the specified id
- deleteCrossing(crossingId):void
  if it exists, it deletes the crossing with the specified id
- laneExists(crossingId, laneId) boolean
  returns true if a lane exitst with the specified Ids
- readLane(crossingId,laneId):Lane
  if it exists, it returns the lane with the specified Ids
- deleteLane(crossingId,laneId):void
  if it exitsts it deletes the lane with the specified Ids
- requestTrainCrossing(crossingId):Request
  returns a request, containing the data about the permission grant/denial
  places a priorityLock on the crossing, no matter the outcome
  the identity of the requester is recorded and stored as a hash
- releaseTrainPermission(crossingId,requestId):void
  if the request and crossing exist, it releases the permission
- requestCarCrossing(crossingId,laneId):Request
  returns a request, containing the data about the permission grant/denial
  the identity of the requester is recorded and stored as a hash
- releaseCarPermission(crossingId,requestId,laneId):void
  if the request, crossing and lane exist, it releases the permission

Actors

- Railway infrastructure managers
- Trains
- Autonomous vehicles

Entities

Railway crossing

id	laneIds	state	prioritylock
string	string	“LOCKED”/“FREE TO CROSS”	boolean

- id is a unique identifier of the crossing
- laneIds are the lanes belonging to the crossing
- state specifies whether the crossing can be crossed freely by the train
- prioritylock is true when a train has requested permission to pass

Lane

id	crossingid	capacity	occupied	prioritylock
string	string	int	int	boolean

- id in combination with the crossingid uniquely identifies the lane
- capacity is the number of cars that can use the lane at the same time
- occupied is the number of cars currently using the lane
- prioritylock is true when a train has requested permission to pass the crossing the lane belongs to.

Request

id	crossingId	laneId	roleOfRequester	granted	active
long	string	string	TRAIN/CAR	boolean	boolean

- id is a random number and in combination with crossingId and laneId it uniquely identifies all requests
- crossingId is the Id of the crossing the request was published to
- laneId is the Id of the lane the request was published to
- roleOfRequester signifies wheter the request was submitted by Train or autonomous vechicle
- granted is true when the permission to cross was granted
- active is true until the permission to cross has been released by the requester

RequestPrivateData

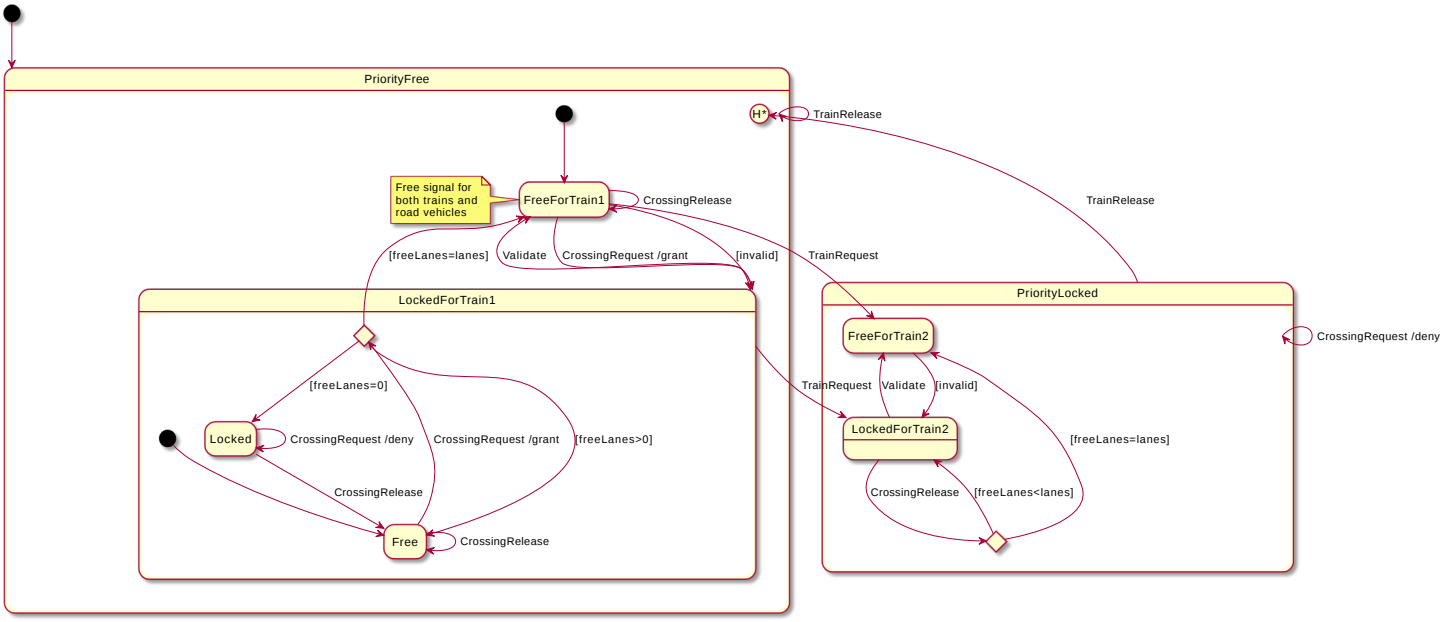
id	laneId	crossingId	clientId
String	String	String	String

- id is the id if the request in String form and along with laneId and crossingId it uniquely identifies the request, that the private data belongs to
- laneId is the id of the lane the request was submitted to
- crossingId is the id of the crossing the request was submitted to
- clientId is the Id of the requester, given by their organization's CA, as this is sensitive information, this is stored as a hash

State Machines

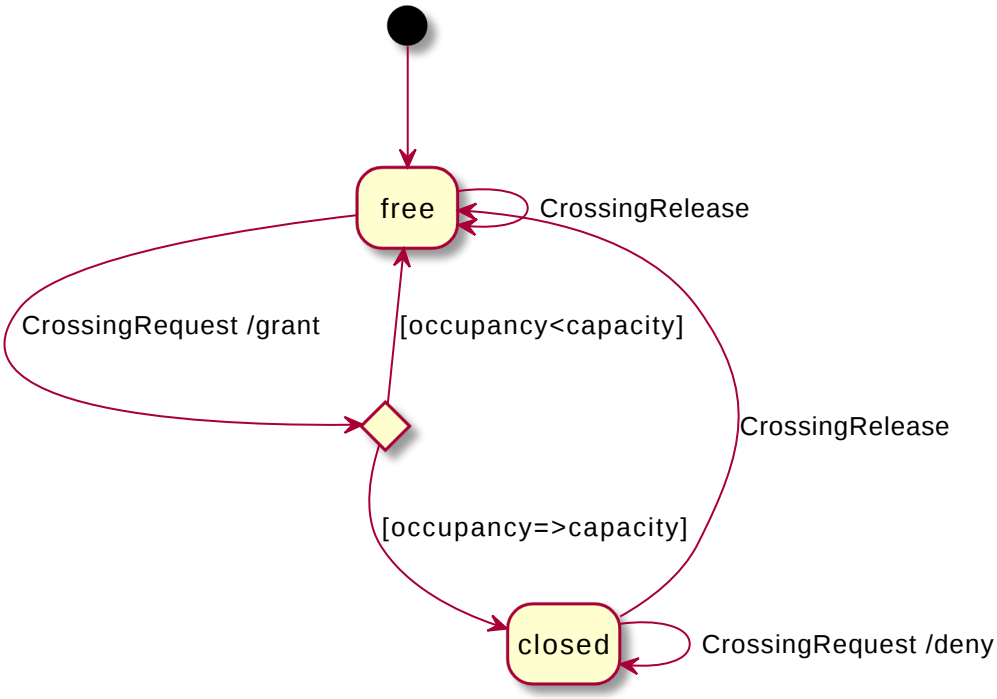
System overview

The diagram below depicts the main behaviour of the system. The periodic refreshments of free crossing validity have been omitted to improve understandability, but after a predetermined duration has passed in any of the FreeForTrain states, the invalid variable becomes true, and the active state transitions to the LockedForTrain states. From there if the Validate signal is triggered the state can transition back to the FreeForTrain state.



Lanes

An intersection can contain one or more lanes, each of which can serve one or more cars at once. The amount is determined by the railroad infrastructure management. The diagram below shows the behaviour of each lane.


free
CrossingRelease
CrossingRequest /grant
[occupancy<capacity]
CrossingRelease
[occupancy=>capacity]
closed
CrossingRequest /deny