

# SFERA Implementation Cases

# List of implementation Cases

## **Train preparation**

- Driver pre-loads the data before the train starts (2 train services) (ex: Loading an S-DAS)

## **Single driver**

- Single driver for single train service without disconnection (proper app quit)
- Single driver for single train service without disconnection (non-proper app quit)
- Single driver for single train service with another person consulting the train journey on another tablet (e.g. instructor)
- Single driver for single train service with lost connection (short)
- Single driver for single train service with lost connection (long)
- Single driver with two successive train services to operate

## **ATO**

- Train service with an ATO over Class B part of the trip




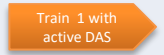






## **Driver change**

- Driver change on single train service (first driver properly disconnects)
- Driver change on single train service (second driver takes over)

## **Border crossing**

- Border crossing
- Driver change at border crossing

# Key

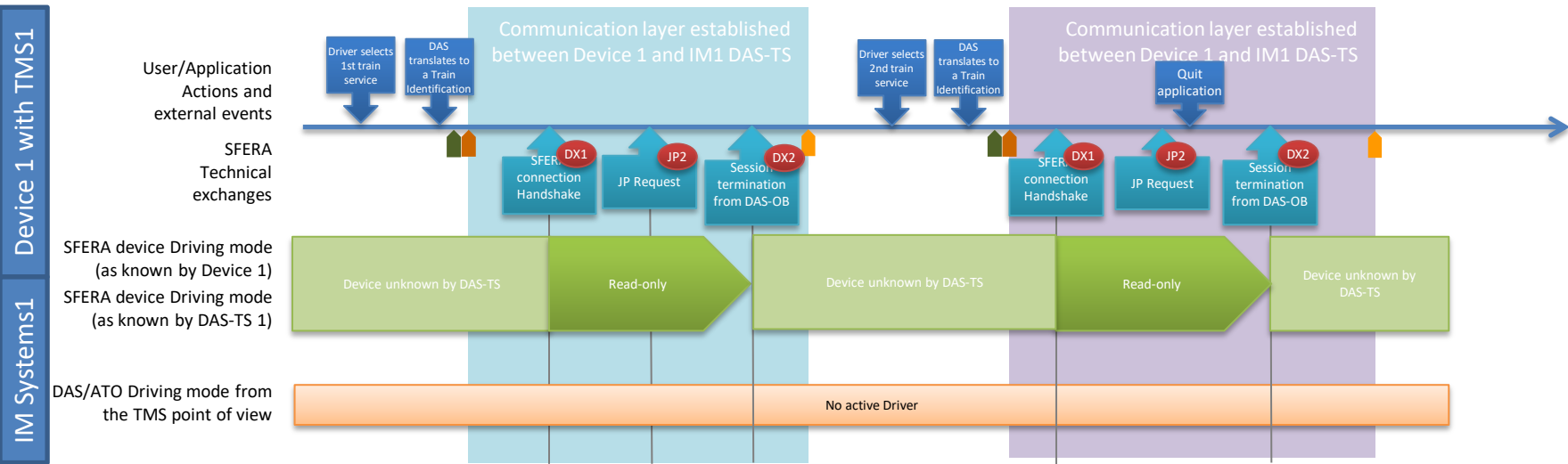
	Graphic element	Definition
General		External events
		Systematic first connection sequence that is described in the first implementation cases
Driving mode		Per Device Driving mode as seen by the DAS-TS and the Device. In cases where the Driving mode considered by both systems is different, two arrows are represented in parallel.
		DAS Driving mode of the train as seen by the TMS. Even if multiple devices are connected to the DAS-TS, the TMS will only consider a unique Driving mode per train.
Use cases		Use case triggered. The implementation cases only reference the first use case triggered by an event. This given use case can in turn trigger other use cases. For example, a JP2 Journey Profile Request can trigger a JP3 and/or JP4 use case.
		Use case reference
Communication layer		Authorisation procedure that results in an authorisation token for a train movement
		Connect primitive to connect on technical level with message broker
		Close primitive to close connection with message broker
		Communication established between TS and OB

# Implementation Cases

Train Preparation

# Driver pre-loads the data before the train service starts

## (Two trains on same IM infrastructure) One JP Request per train service



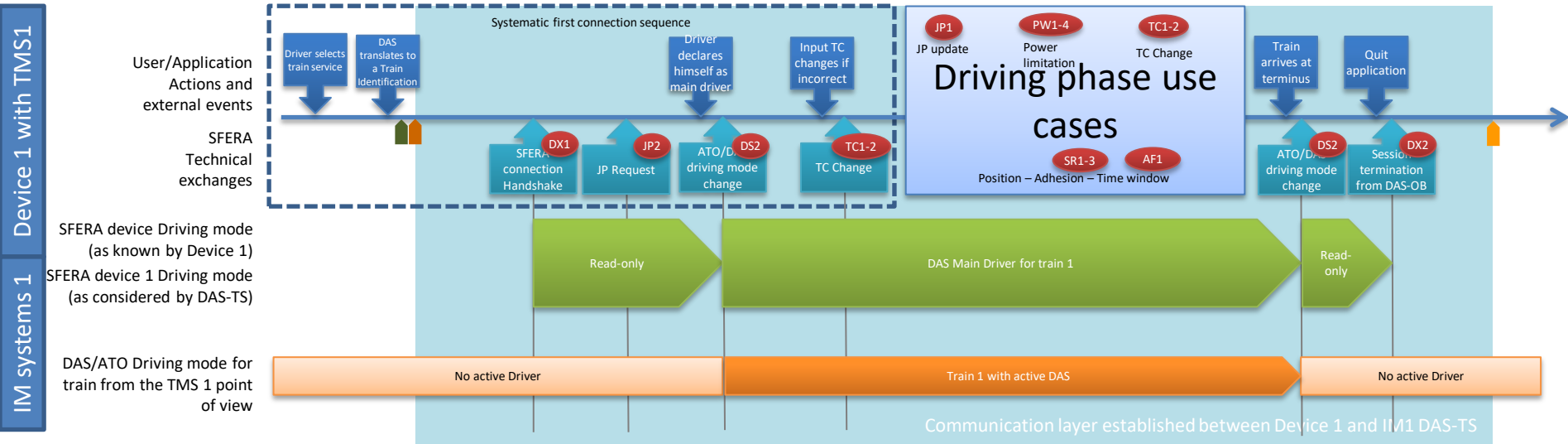
The token references the RU, and a train for which the device is authorised: A SFERA connection between a device and the DAS-TS opens the possibility for the device to access information on a specific train of the RU. It is the RU's responsibility to limit the requests of the device to the trains for which the driver is authorised.

Although this implementation case presents the two separate sequences being established and closed sequentially, the protocol permits the device to keep the communication layer open and start the second JP Request sequence with a newly requested token for the second train.

# Implementation Cases

Single Driver

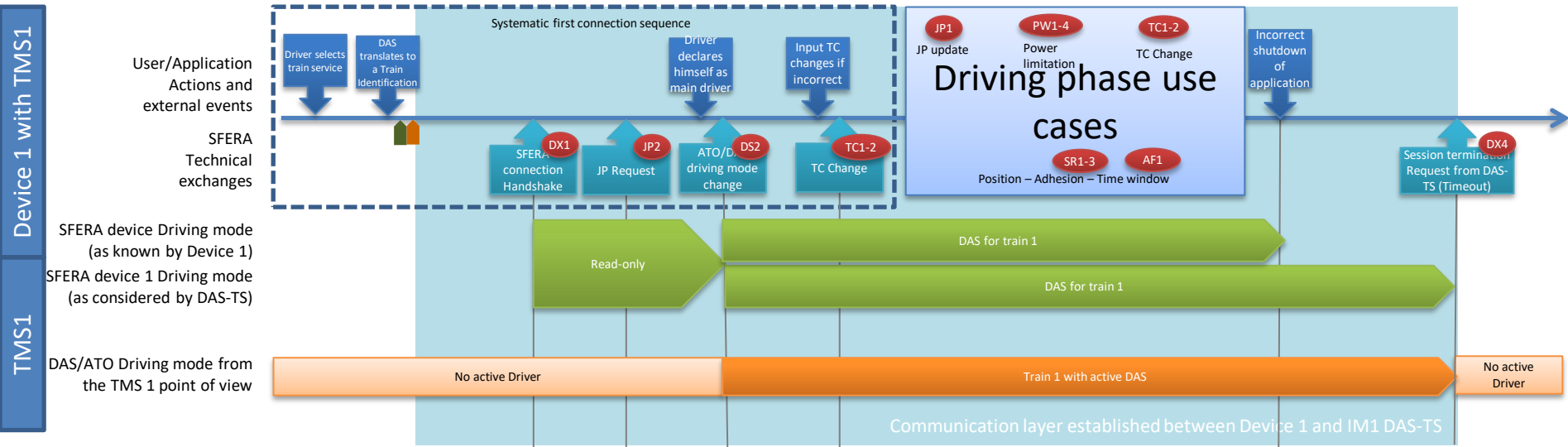
# Single driver for a single train service without disconnection (proper app quit)



Alternatives for the implementation cases:

- The driving mode change when the train arrives at the terminus could be omitted.
- If the DAS is authorised only for this train, the disconnection could be automatically done by the device, without going through the « read-only » phase.

# Single driver for a single train service without disconnection (non proper app quit)

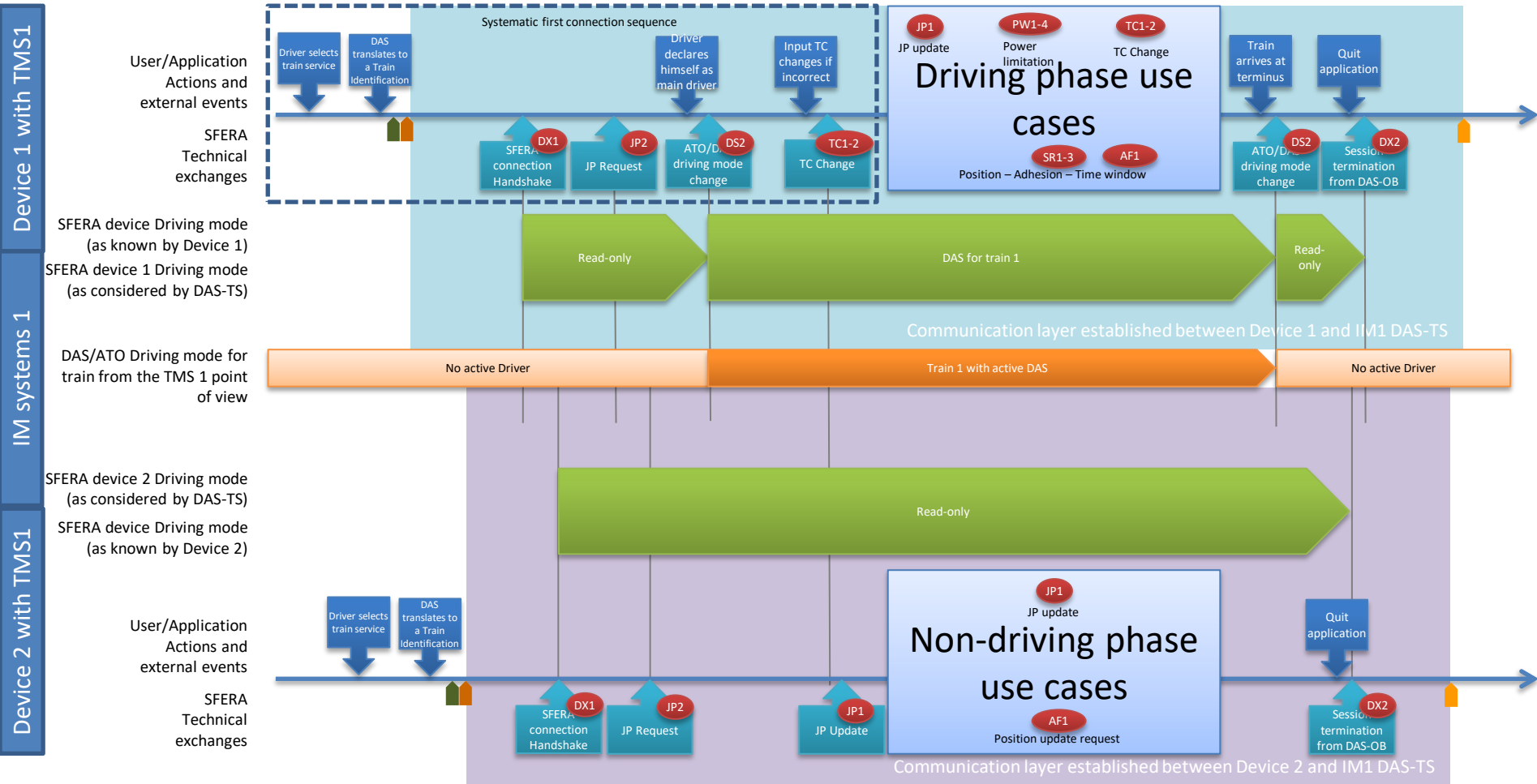


The connection could also be closed by the IM before the timeout if:

- The authorisation token expires (IM disconnects the device)
- An invalid or no token is sent with a request/event message

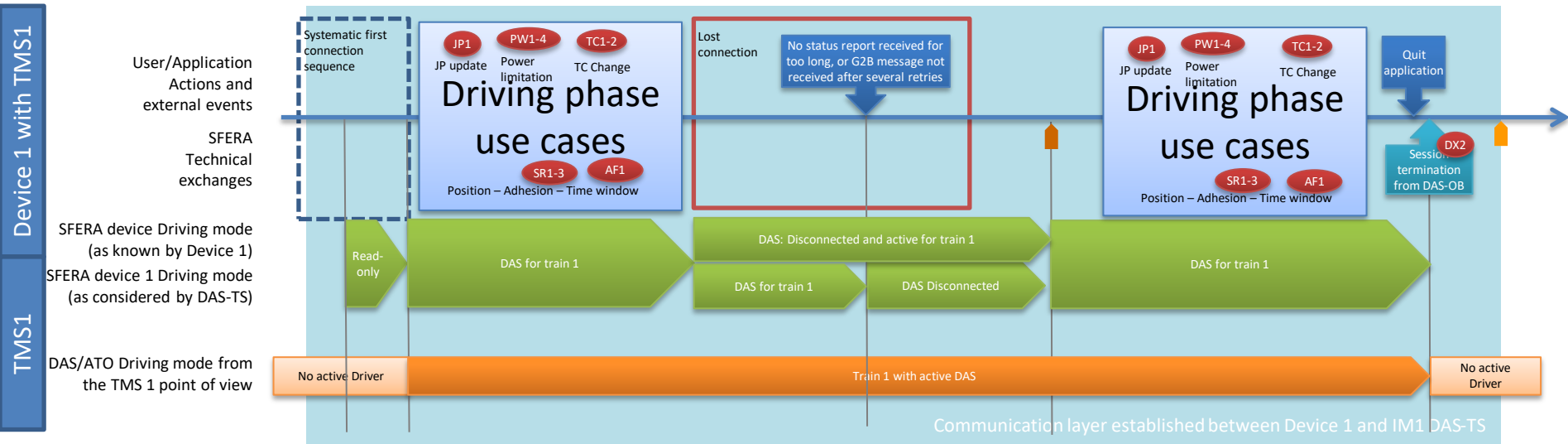


# Single driver for a single train service with another person consulting the train journey on another device (e.g. instructor)



A TC change from the active driver will result in the creation of a new TC in DAS-TS, sent to all devices following that train service (via the JP updates).

# Single driver for a single train service with lost connectivity (short)

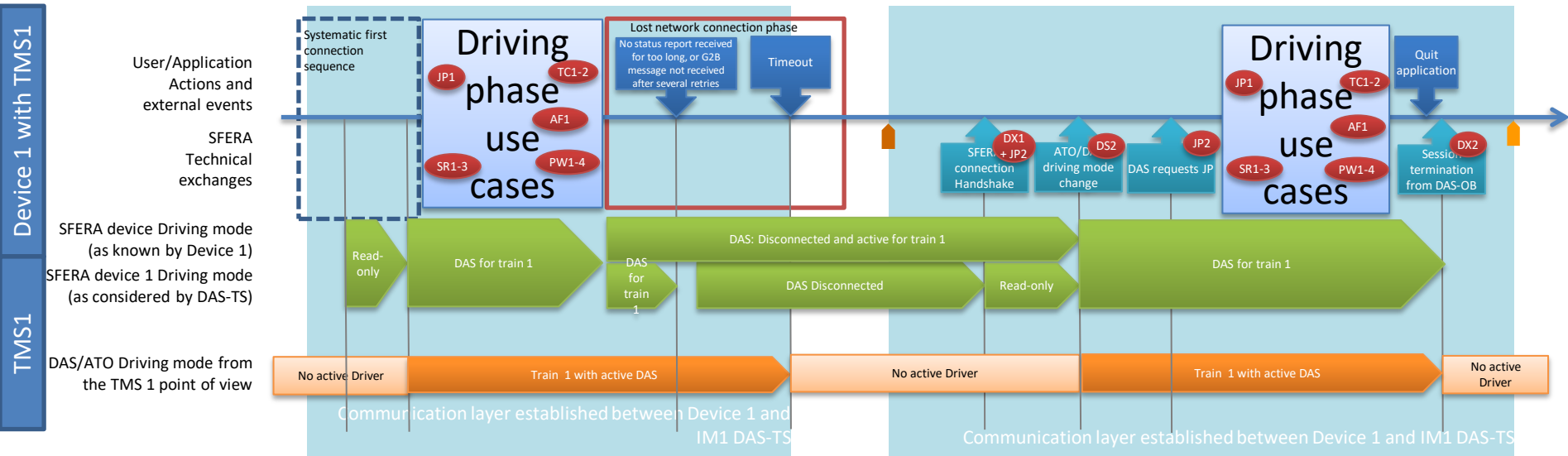


The communication layer does not have a specified requirement concerning immediate detection of a lost connection. A disconnection is therefore only first detected when the DAS-TS does not receive a status report in the specified time frame.

The communication protocol implements a guaranteed delivery requirement. Upon radio reconnection, the device will receive, depending on the parameters used for MQTT:

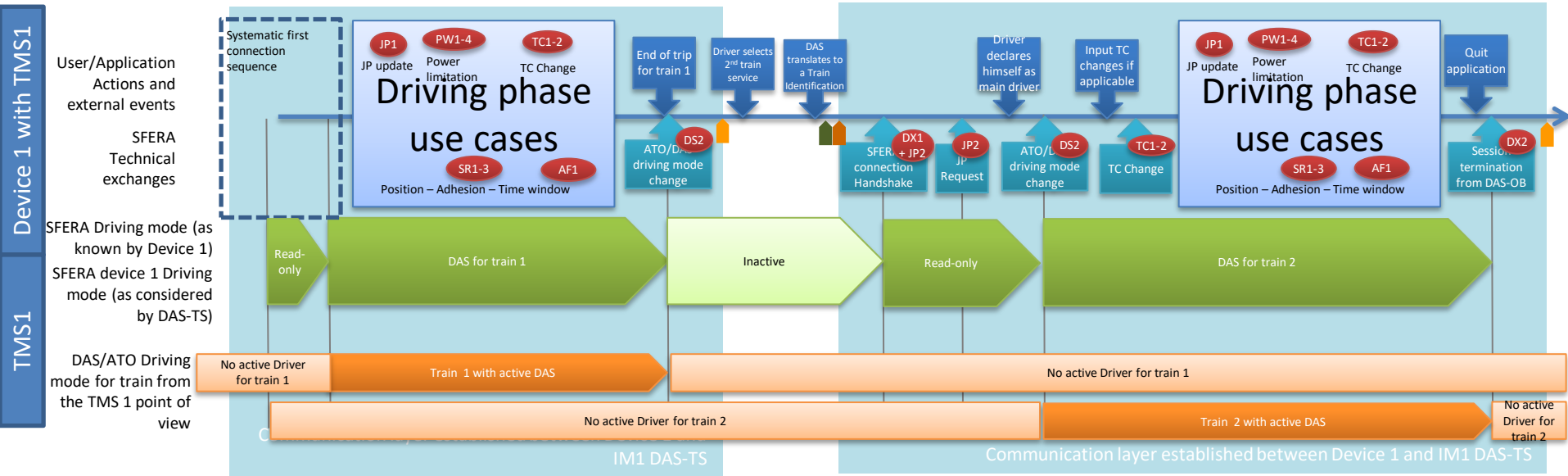
- all the JP updates generated by the TMS while the radio connection was lost;
- the last JP update sent.

# Single driver for a single train service with lost connectivity (long)



Decisions on the time value for the timeout and on the functional actions upon timeout are left to each IM. They are exchanged during the handshake. Upon reconnection, the device will automatically reset a connection without repeating the « systematic first connection sequence ». The driver therefore is not required to take action (no train number entry, etc.).

# Single driver with two successive train services to operate

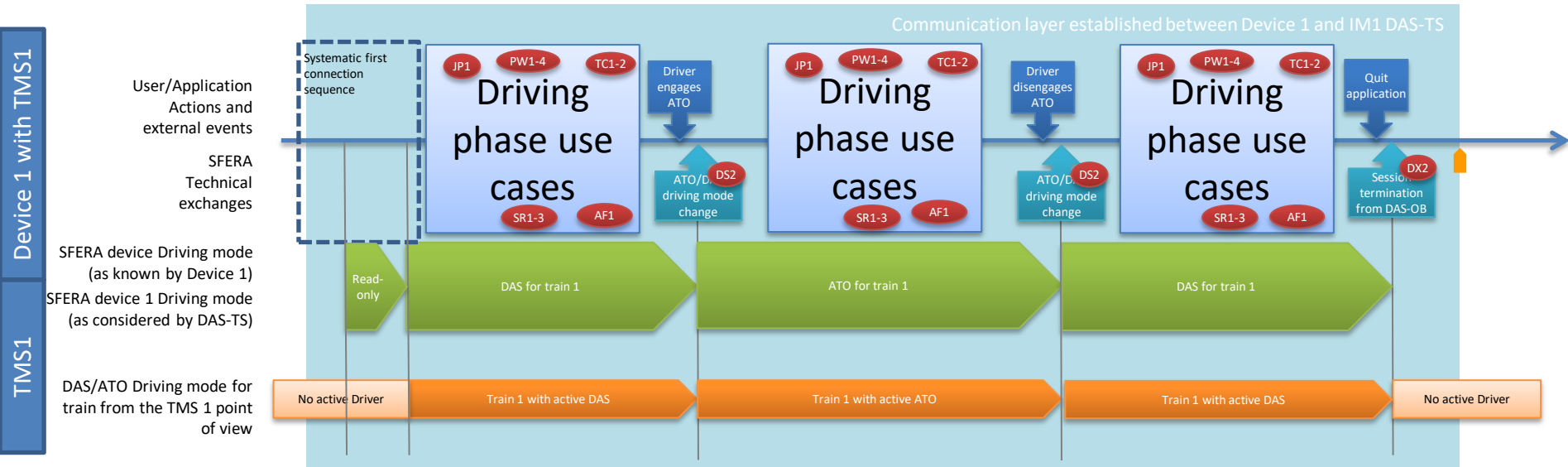


The standard practice for SFERA would be to have a different token for each service. Therefore, a new token will need to be generated in the connection exchanges for the 2<sup>nd</sup> train.

# Implementation Cases

Interactions with ATO

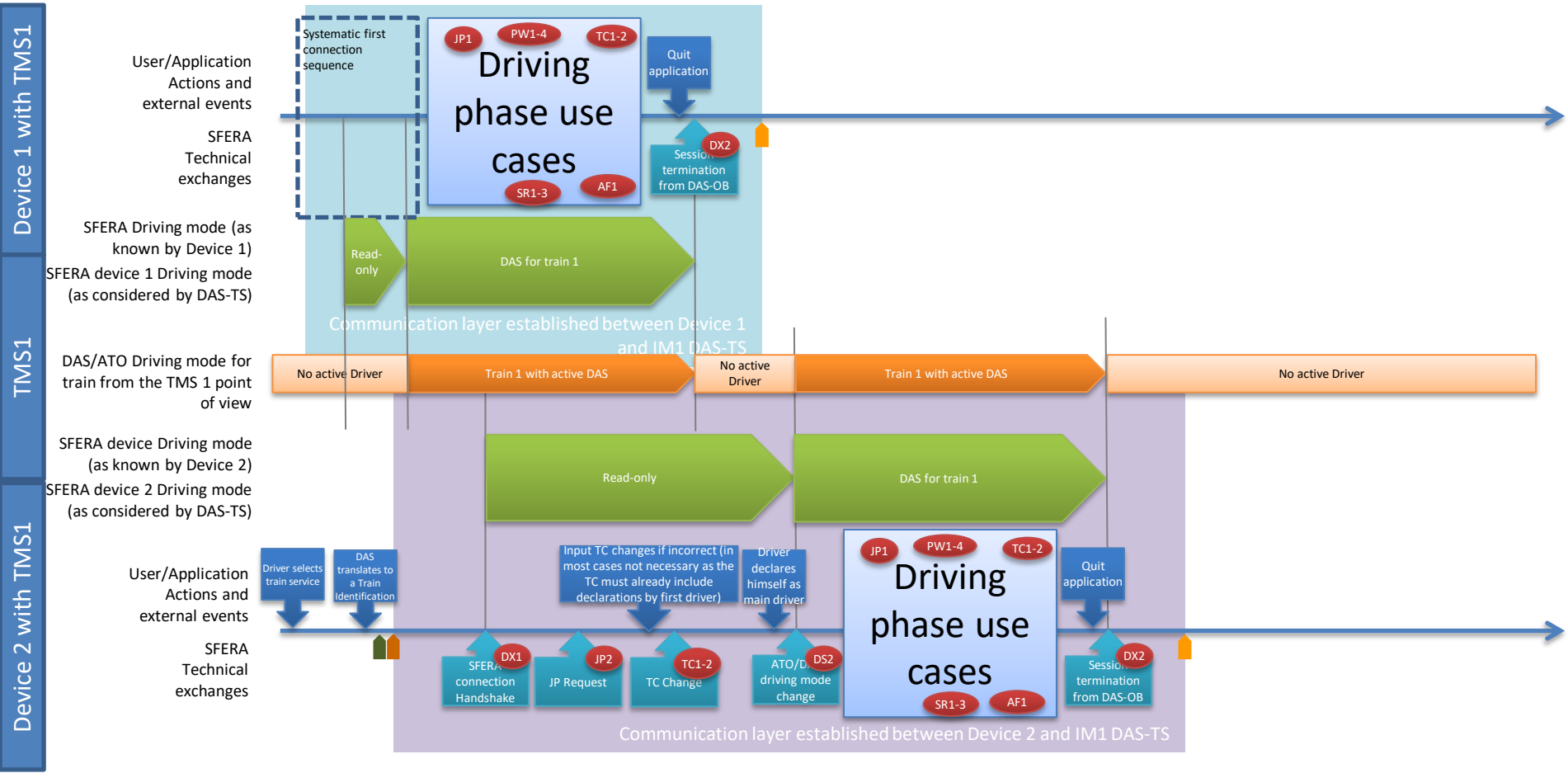
# Train service with an ATO over Class B part of the travel (same device)



# Implementation Cases

Driver Change

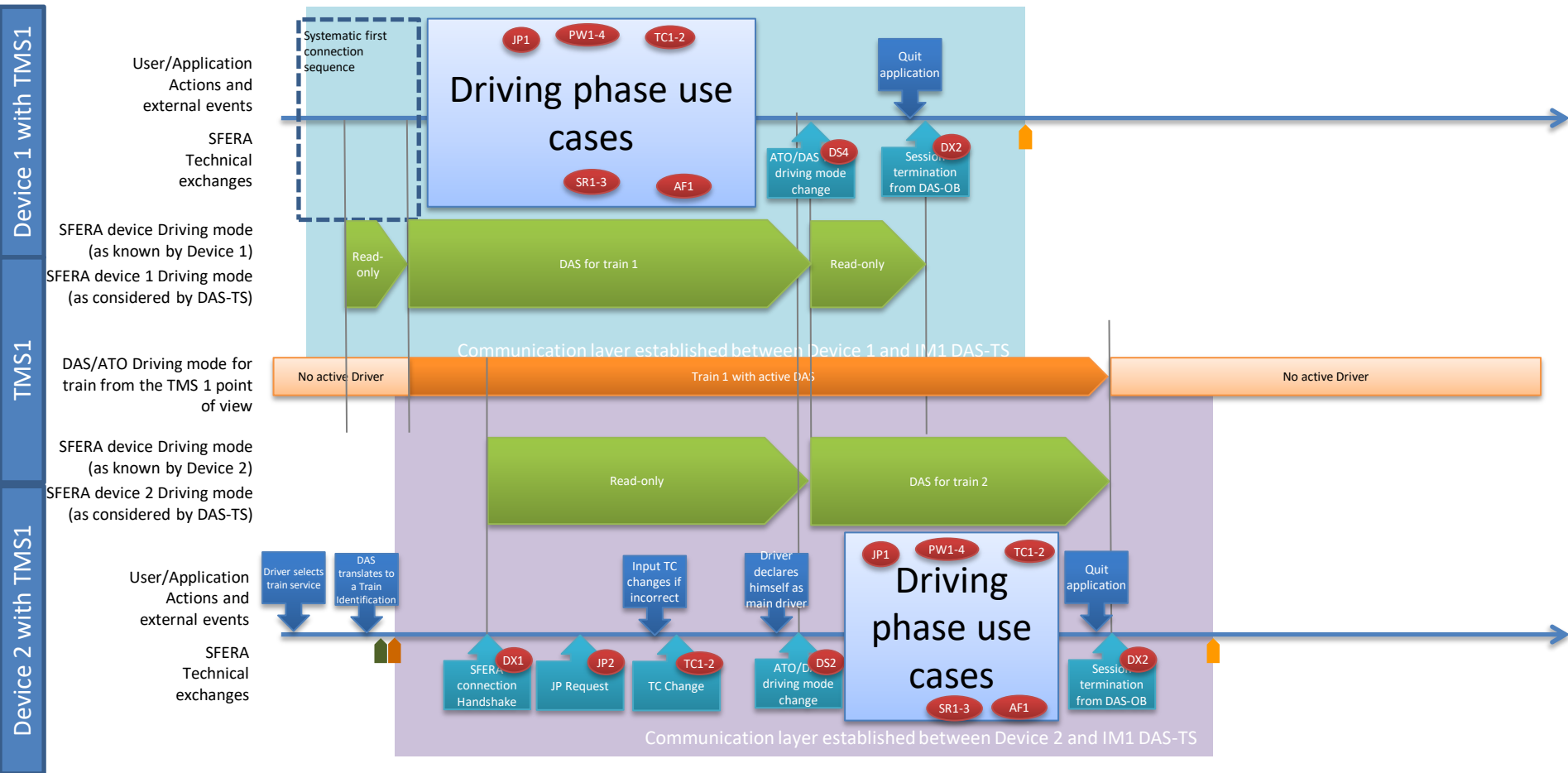
# Driver change on a single train service (first driver properly disconnects)



During the phase where Device 2 is in read-only mode and Device 1 is active, Device 2 will receive all JP updates, and will therefore have received all TC changes declared by the driver with Device 1.



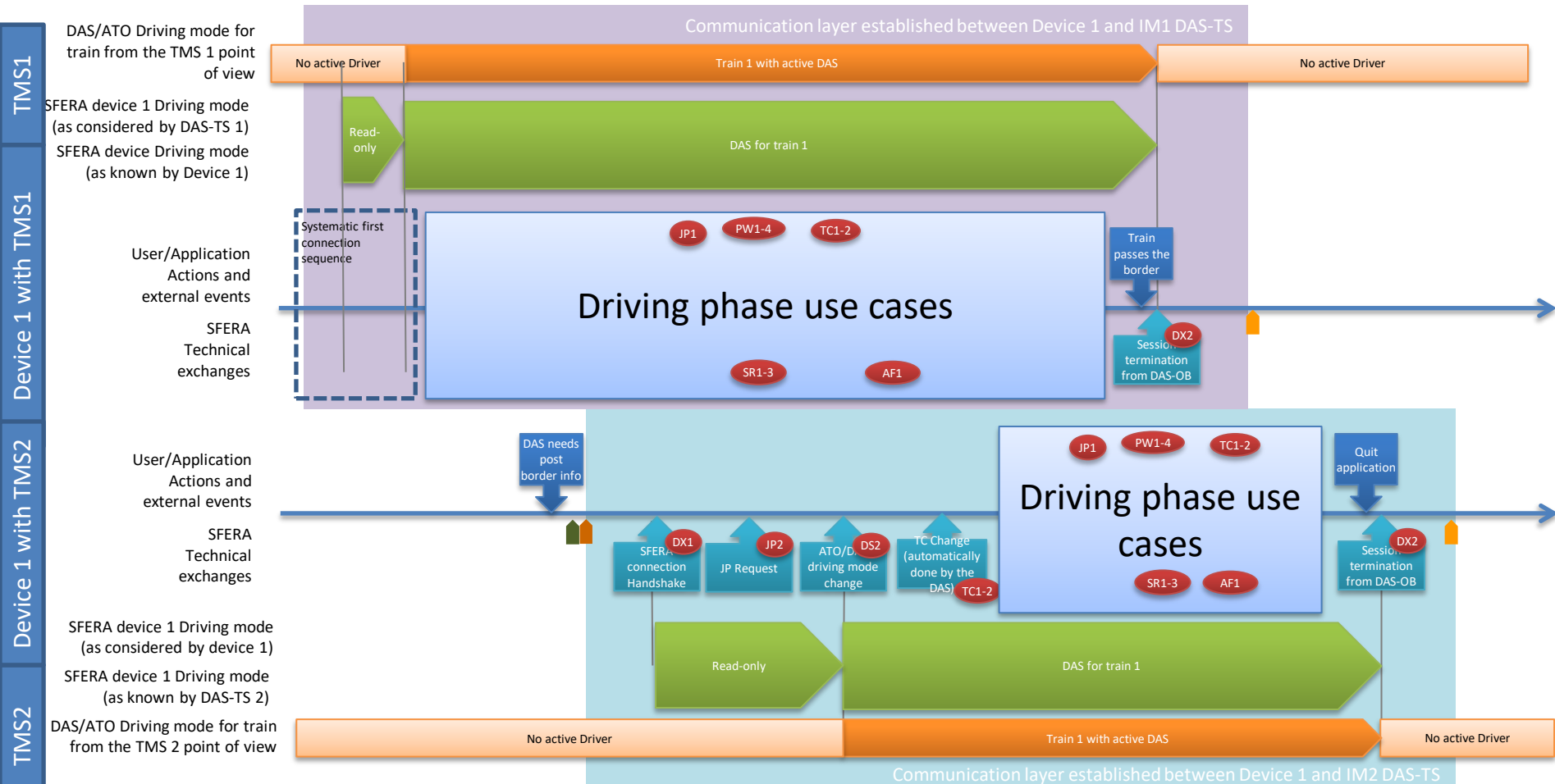
# Driver change on a single train service (second driver takes over)



# Implementation Cases

Border Crossing

# Border crossing



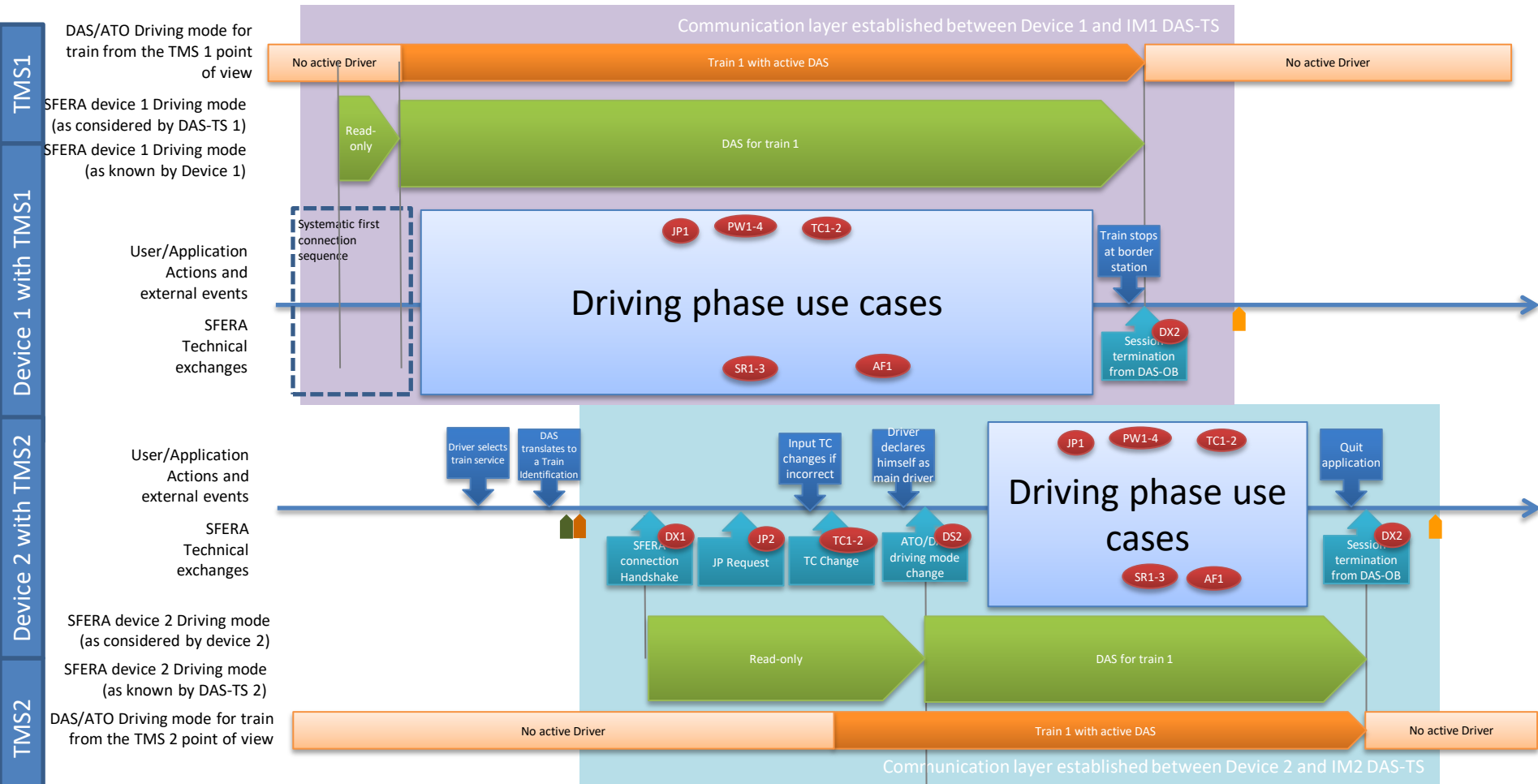
The DAS tries to connect to TMS2 as soon as it considers that it needs data after the border point. This will certainly be at least before the last timing point before the border, but can be much earlier (the TMS2 connection could be initiated just after the TMS1 connection).

The border crossing can be concomitant with an RU change, without a device change : the communication to TMS 2 is opened with a different token based on the second RU's key.

This implementation applies to a hand-held device or the integrated DAS of a rolling stock element

The two RUs have to agree on the conditions in which their respective datasets are handled from a confidentiality point of view (for example the possibility for the same application on Device 1 to process data for both RUs at the same time ).

# Border crossing with a driver change (person-bound devices)



This implementation case assumes that Device 1 knows that it must not connect to the next IM (information not included in the SFERA protocol). If not, the transition described in Slides 12 and 13 is used.