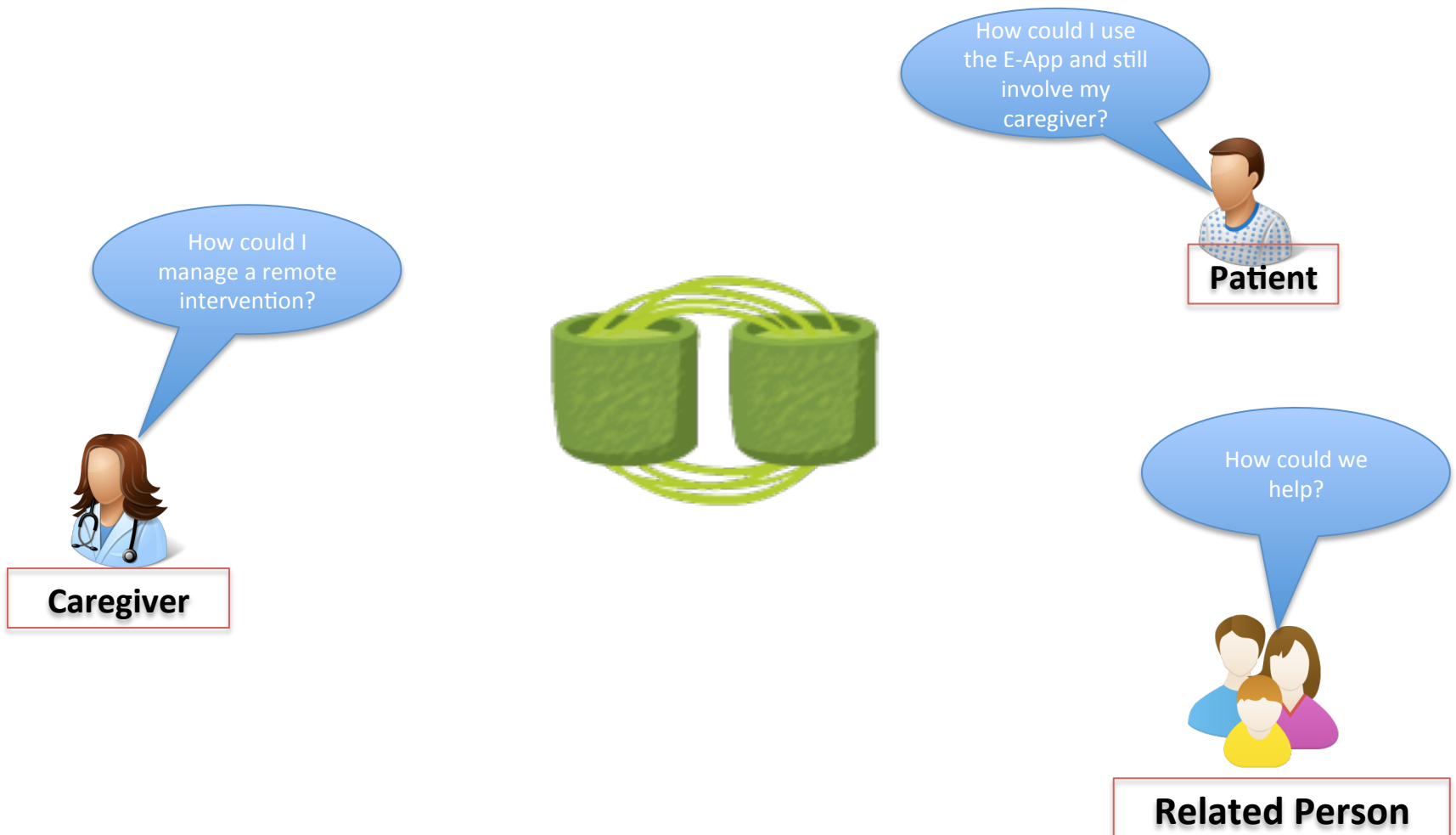


# KickAss Functional Flow

Architecture team

# Functional scope



# Functional flow

- Kick Ass Process flow:
  - Start the game :
    - Authenticate
    - Start the game
  - During the game:
    - Update the state: from CarePlan to the Game and from the Game to the CarePlan and ActivityStatus
    - Update the subactivities

# 1. Authentication en Start Game

1. Oauth Sequence when starting the game includes the: JS Client Adapter authentication:  
`Koppeltaal.Client.convertAccessCodeToToken`

2. Read Game Data to be able to init the game sequence:

a) `Koppeltaal.Client.getMessageHeaders` for `CreateOrUpdateCarePlan` messages

- sort the message structure of the highest id en get this message from Koppeltaal Server

(`Koppeltaal.Client.getMessageWithId`)

- the highest id should have a correct `CarePlan` and get the main data from it:: `careplan id/ref/status`, `activity id/ref/status/subactivities`, `patient id/ref/name`, `practitioner id/ref/name`. Also check the user ref from the Oauth authorization launch sequence

- If no `CarePlan` is found then the Game cannot start!

3) Read the game status using the `Koppeltaal.Client.getMessageHeaders` for `UpdateCarePlanActivityStatus` messages

- sort the message structure of the highest id en get this message from Koppeltaal Server

(`Koppeltaal.Client.getMessageWithId`)

- the highest id should have a correct `Message` and get the main data from it:: `activity status id/ref/patient`, `activity ref/status/progress/subactivities`, `gamestate id/ref/data`

- the game state data is modified in de game

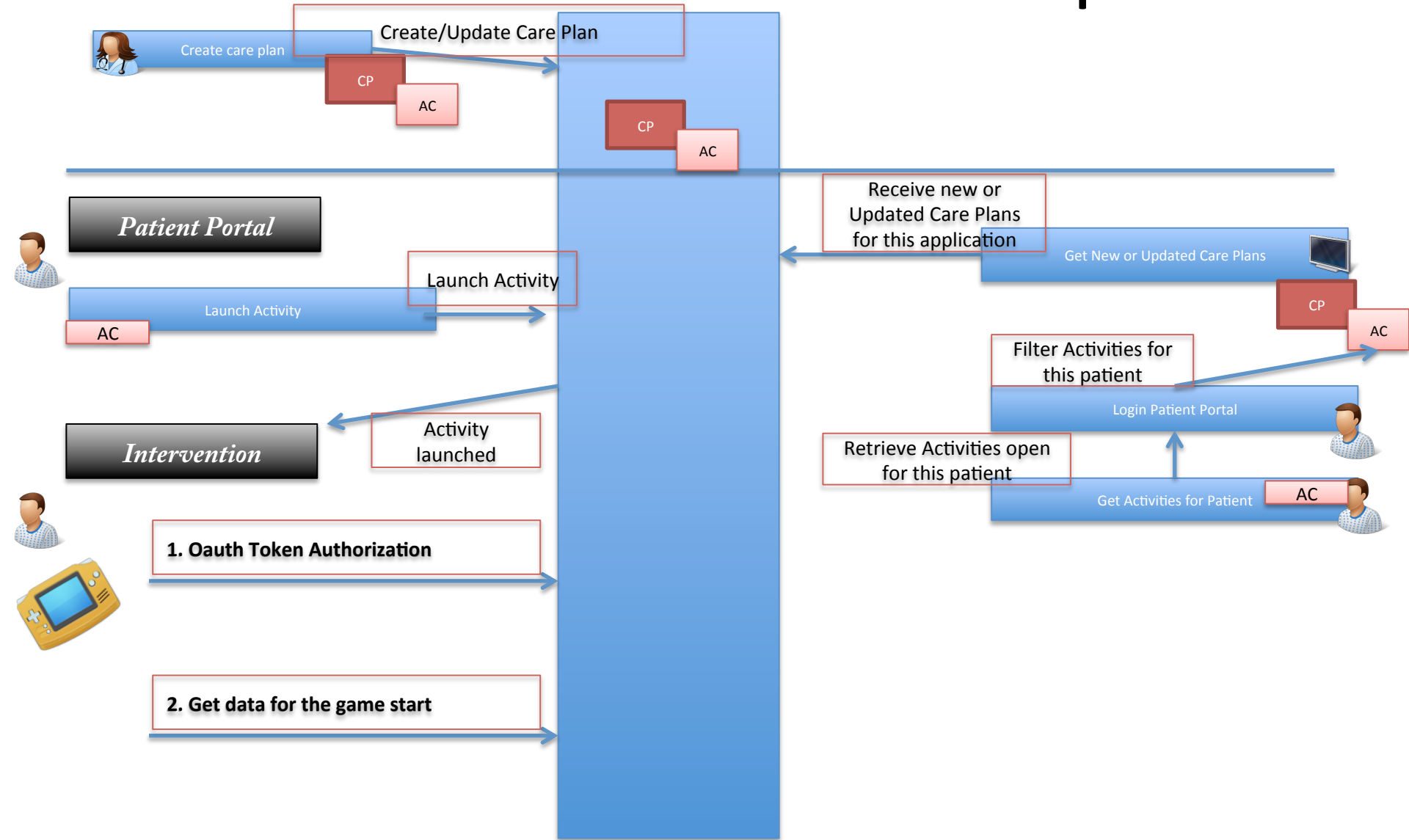
- de `status/progress/subactivities` are not updated in de game but overwritten with the latest game data at any update.

Not Available:

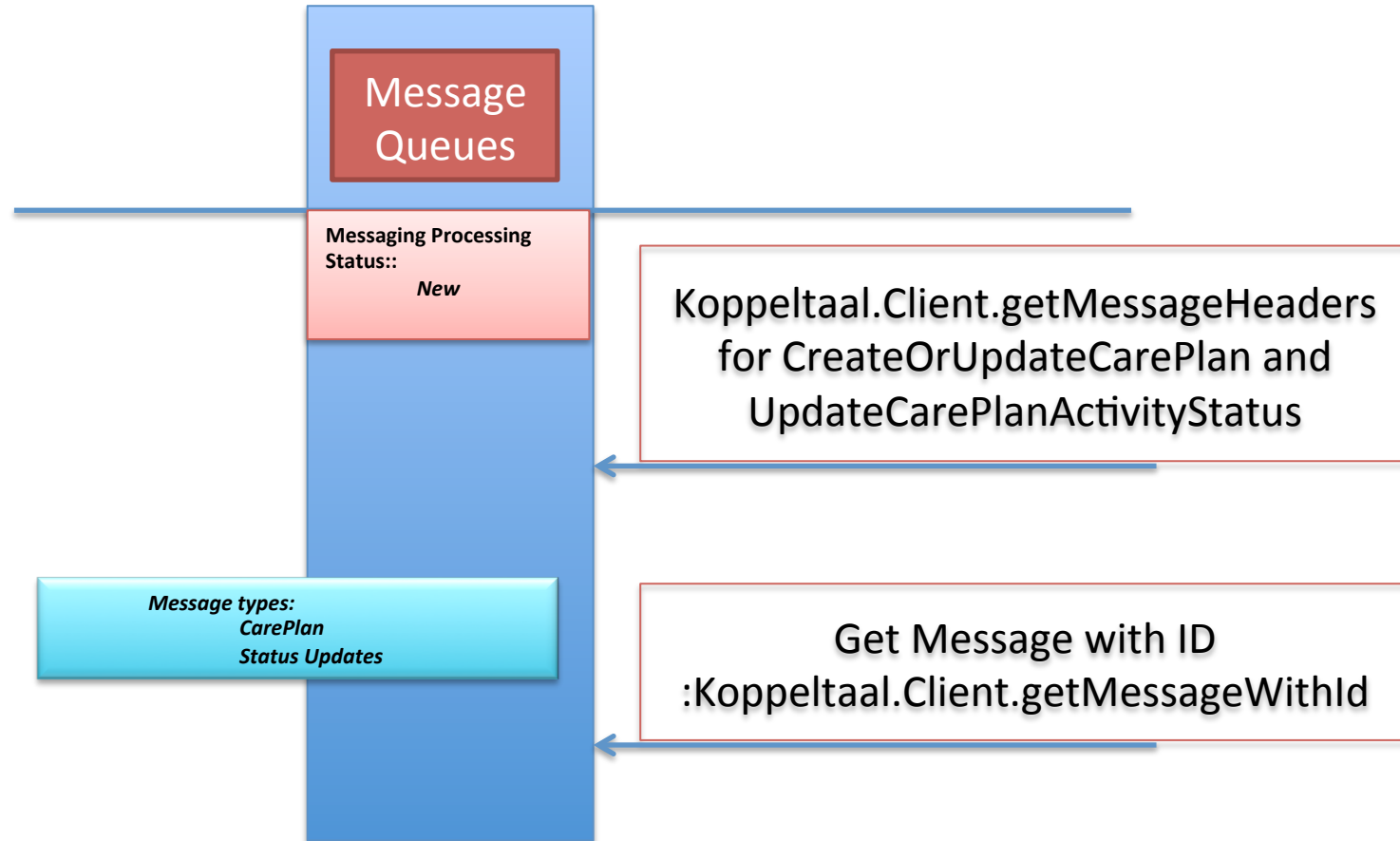
- Related Person flow

- Subactivities are not updated in the game but overwritten with the last game status data

# 1. Process flow – Launch sequence



# 1. API – Get Messages



## 2. Game operations

### 1. Every step in the game

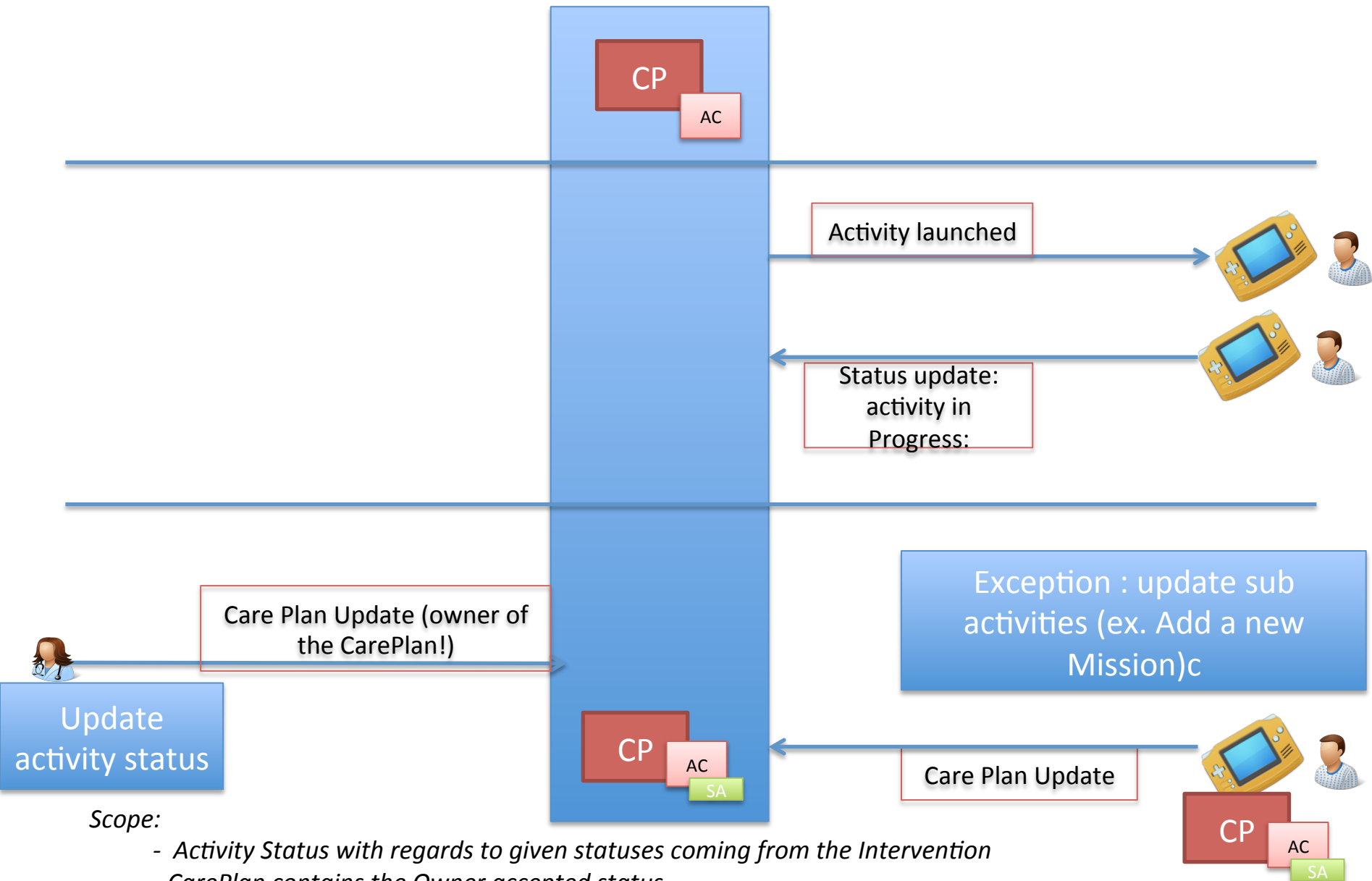
- Get game Data (with a cooldown of 10 sec) as describe before

### 2. Sending of Changes in the Game Data (every time when changes happen in the Game):

Koppeltaal.Client.postMessageToMailbox voor UpdateCarePlanActivityStatus message

- Latest game data stored in a blackbox
- Activity status/progress/subactivities are equal with the game data
- If no UpdateCarePlanActivityStatus was created the one is made
- If the de subactivity ids in the game are different then the ones received in the CarePlan then Koppeltaal.Client.postMessageToMailbox for CreateOrUpdateCarePlan message (the new game subactivity ids are defined in the message)

## 2. Process flow – Activity status update



### Scope:

- Activity Status with regards to given statuses coming from the Intervention
- CarePlan contains the Owner accepted status
- Owner of the CarePlan should be the only one that can change it (!!)



# 2. Process flow – Activity status update

