CERNHrLifecycleHandler

retrieve **experimentName** from **CernProperties.experiment** configuration value

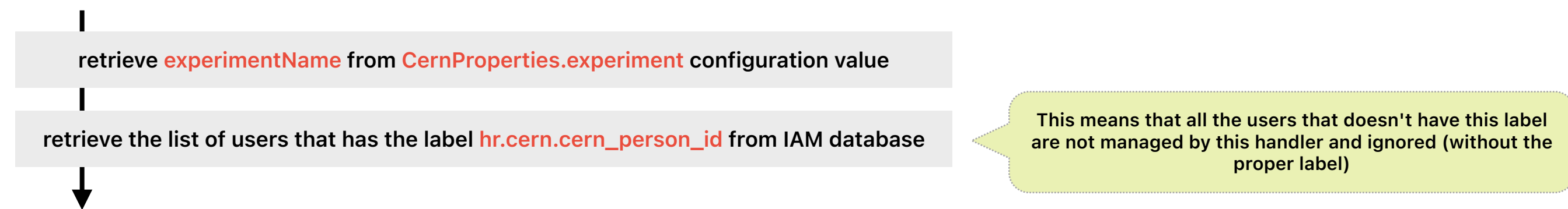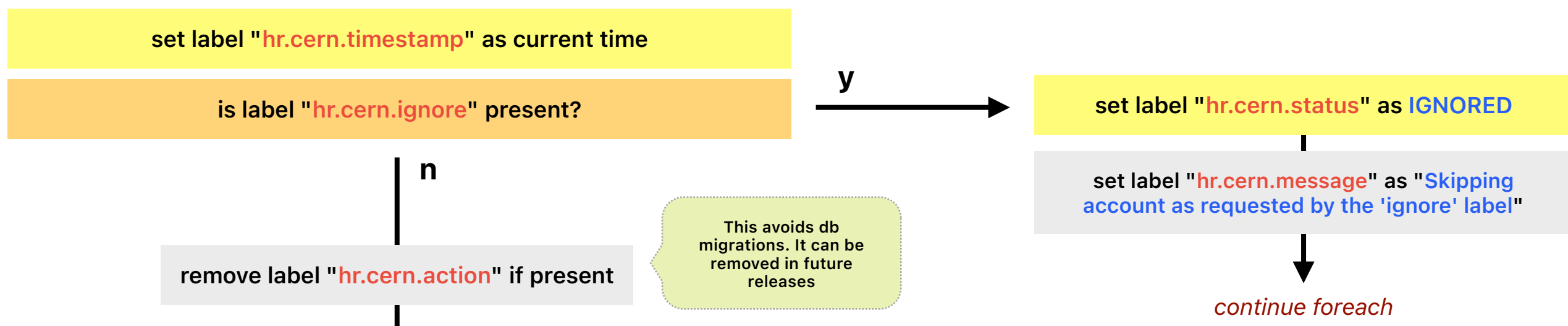retrieve the list of users that has the label **hr.cern.cern_person_id** from IAM database

> This means that all the users that doesn't have this label are not managed by this handler and ignored (without the proper label)

foreach **IamAccount** *that has a "hr.cern.cern_person_id" label*

set label "**hr.cern.timestamp**" as current time

is label "**hr.cern.ignore**" present? — **y** → set label "**hr.cern.status**" as IGNORED

set label "**hr.cern.message**" as "Skipping account as requested by the 'ignore' label"

continue foreach

n

remove label "**hr.cern.action**" if present

> This avoids db migrations. It can be removed in future releases

HR DB API
hr-db-api.web.cern.ch

retrieve user's **VOPerson** record — **RuntimeException** or null **VOPerson** → set label "**hr.cern.status**" as ERROR

set label "**hr.cern.message**" as "Account not updated: HR DB error"

continue foreach

> We don't touch end-time so we totally rely on a manual intervention to eventually extend user's membership when needed

> User info synchronization with VOPerson record retrieved from HR database

Synch **IamAccount** Given Name and Family Name with **VOPerson** record info

is label "**hr.cern.skip-email-synch**" present?
- **y**
- **n** → Does any other account with same **VOPerson** email and a **different UUID** exist??
  - **y** → Log ERROR
  - **n** → Synch **IamAccount** email with **VOPerson** record info

> Search if **experimentName** is included into the **VOPerson** participation list

Has the user a participation on **experimentName**? — **n** → Is "**hr.cern.status**" != ERROR ? — **n** → *continue foreach*

y

set label "**hr.cern.status**" as ERROR

Update **IamAccount** end-time with **current date time**

set label "**hr.cern.message**" as "Account end-time not updated: no participation to experiment found"

continue foreach

> The idea is to delegate to the **ExpiredAccountsHandler** the management of expired users suspension, considering the configured grace periods

Update **IamAccount** end-time with **VOPerson** experiment participation endDate

> We set end-time to the current instant so the user will be set as PENDING_SUSPENSION next time the **ExpiredAccountsHandler** runs

> "participation to the experiment valid" means user's participation to the experiment **endDate** null or greater+equal to current Date, excluding time part

user has a **valid participation to the experiment** ?
- **y** → set label "**hr.cern.status**" as MEMBER
- **n** → set label "**hr.cern.status**" as EXPIRED

set label "**hr.cern.message**" as "Account participation to the experiment is expired"

continue foreach

> No suspension is done. Delegate **ExpiredAccountsHandler** to manage expired users suspension, considering the configured grace periods

Is user **suspended**?
- **y** → set label "**hr.cern.message**" as "Account has a valid participation to the experiment"

  continue foreach
- **n** → User was **suspended** by us?
  - **y** → Restore **IamAccount** (set active **true** + publish event **AccountRestoredEvent**)

    Remove "**lifecycle.status**" label

    set label "**hr.cern.message**" as "Account restored on 2024-09-09 17:32:12"

    continue foreach
  - **n** → *continue foreach*

> To understand if user was suspended by IAM lifecycle handler (and not by an admin) we need to check the existence of label "**lifecycle.status**" + its value equals to PENDING_REMOVAL or SUSPENDED

> Leave users not suspended by IAM internal logic (based on endTime) as suspended

> User can be suspended by admins through the dashboard that means via SCIM API and the account simply is updated by setting active = false, no end-time is touched or label added!

end foreach

> label "**hr.cern.ignored**" is used to exclude a user from updating his end-time with the experiment participation endDate retrieved from HR database. This means administrators have to manually manage his membership endTime to avoid suspension
>
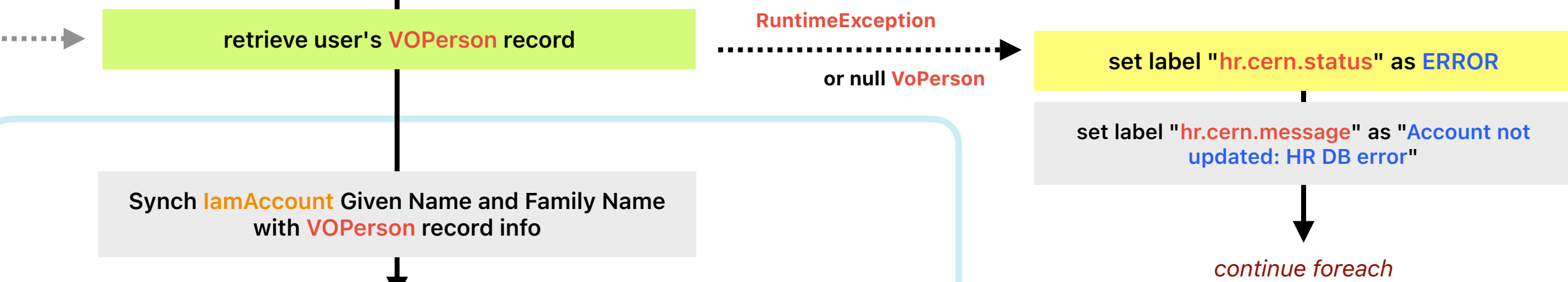> label "**hr.cern.timestamp**" value is the last time HR handler has run
>
> label "**hr.cern.status**" values are:
> - IGNORED when label "**hr.cern.ignore**" is present,
> - ERROR in case of failed connections to HR API or null VOPerson record
> - MEMBER: when API returns the VOPerson record and the participation to the experiment is found and not expired,
> - EXPIRED: when API returns the VOPerson record and the participation to the experiment is found but it's expired
>
> label "**hr.cern.message**" contains a detailed explanation of the status
>
> label "**hr.cern.skip-email-synch**" if set is used to skip email update from HR record (manually set via API)
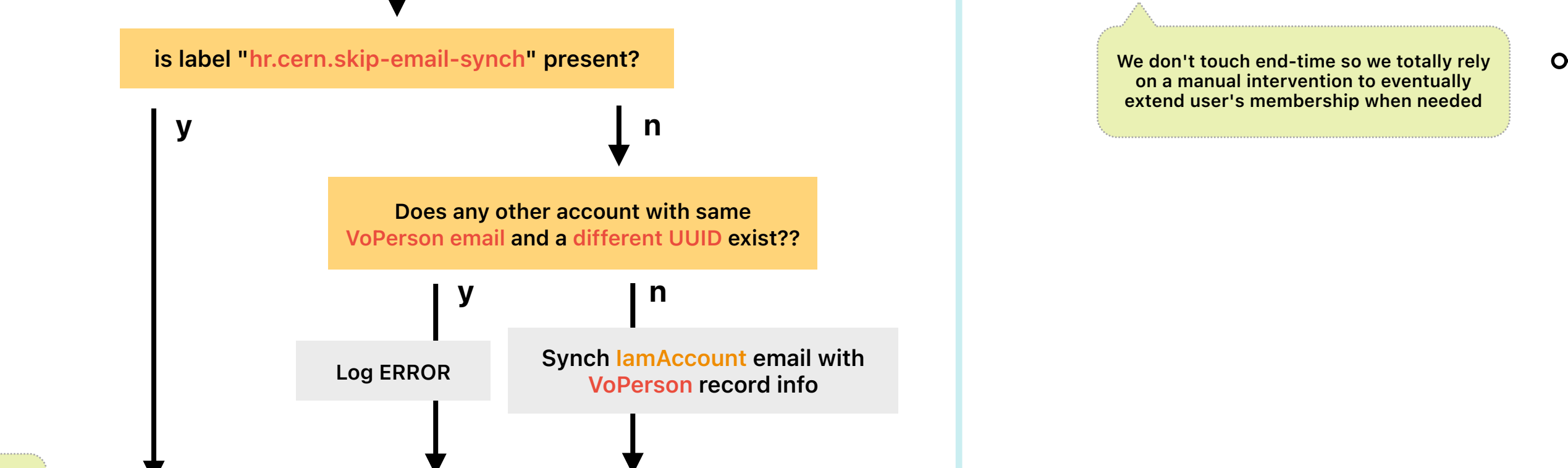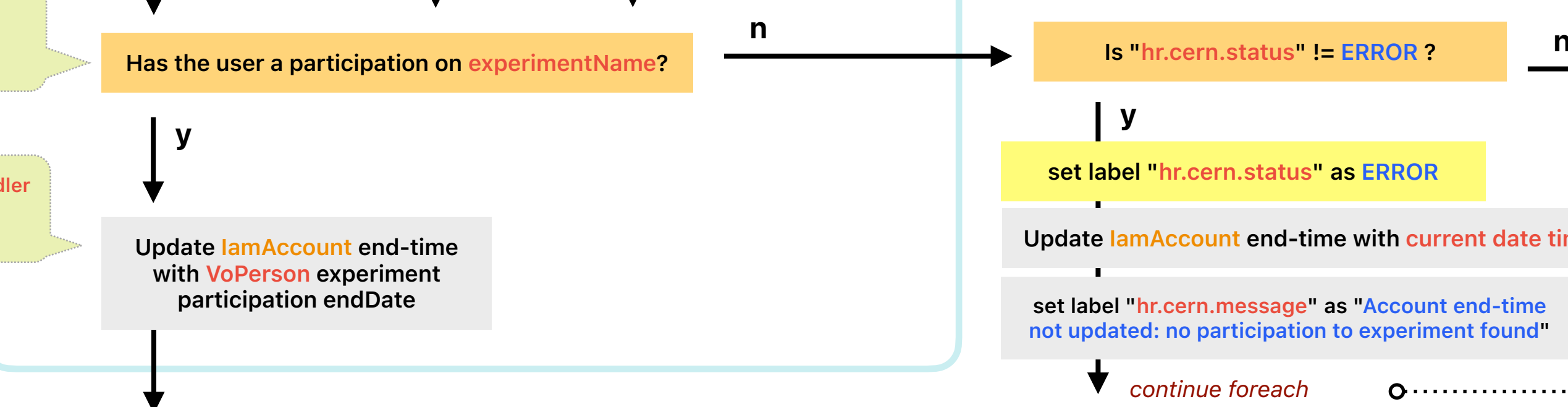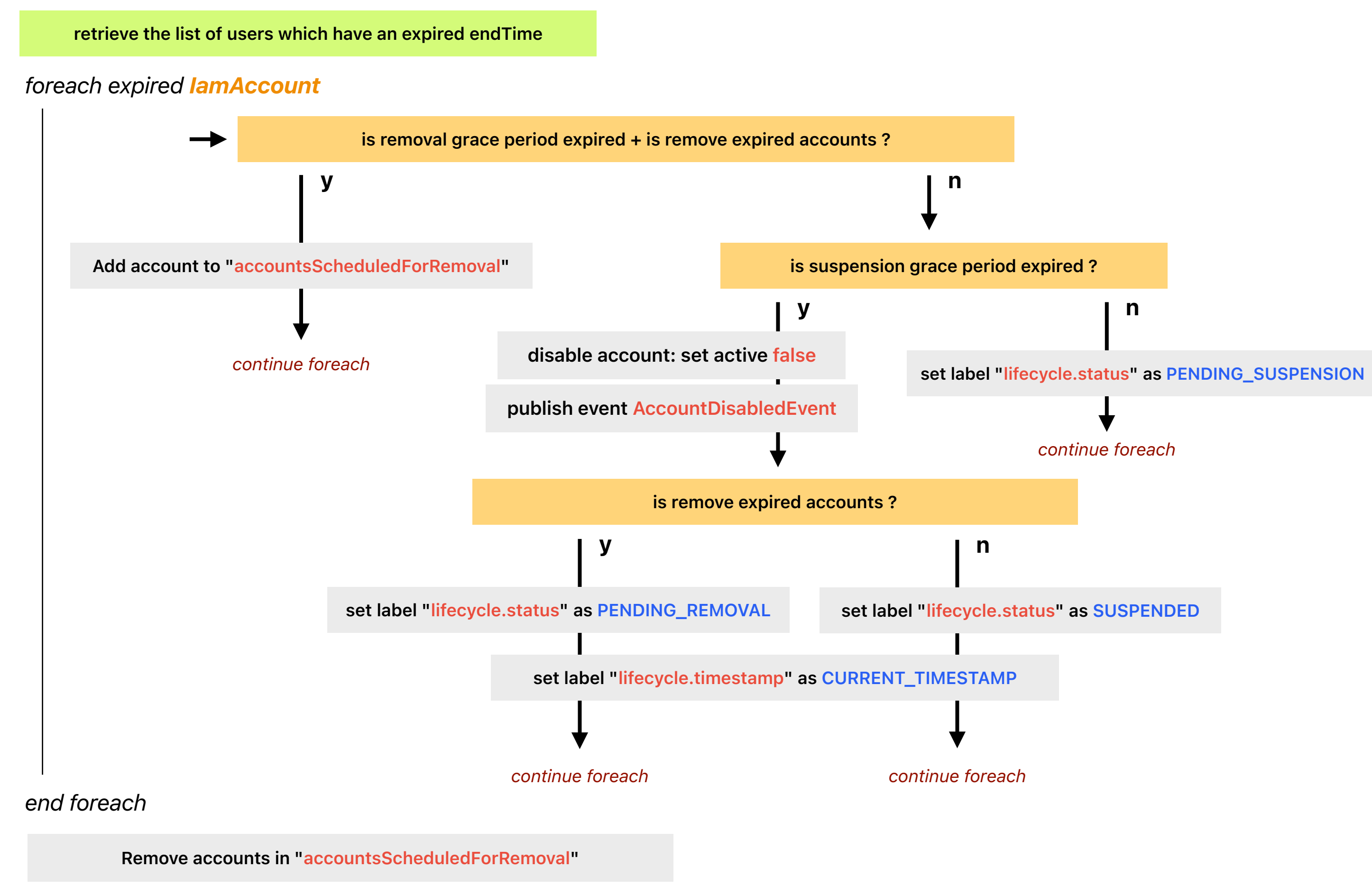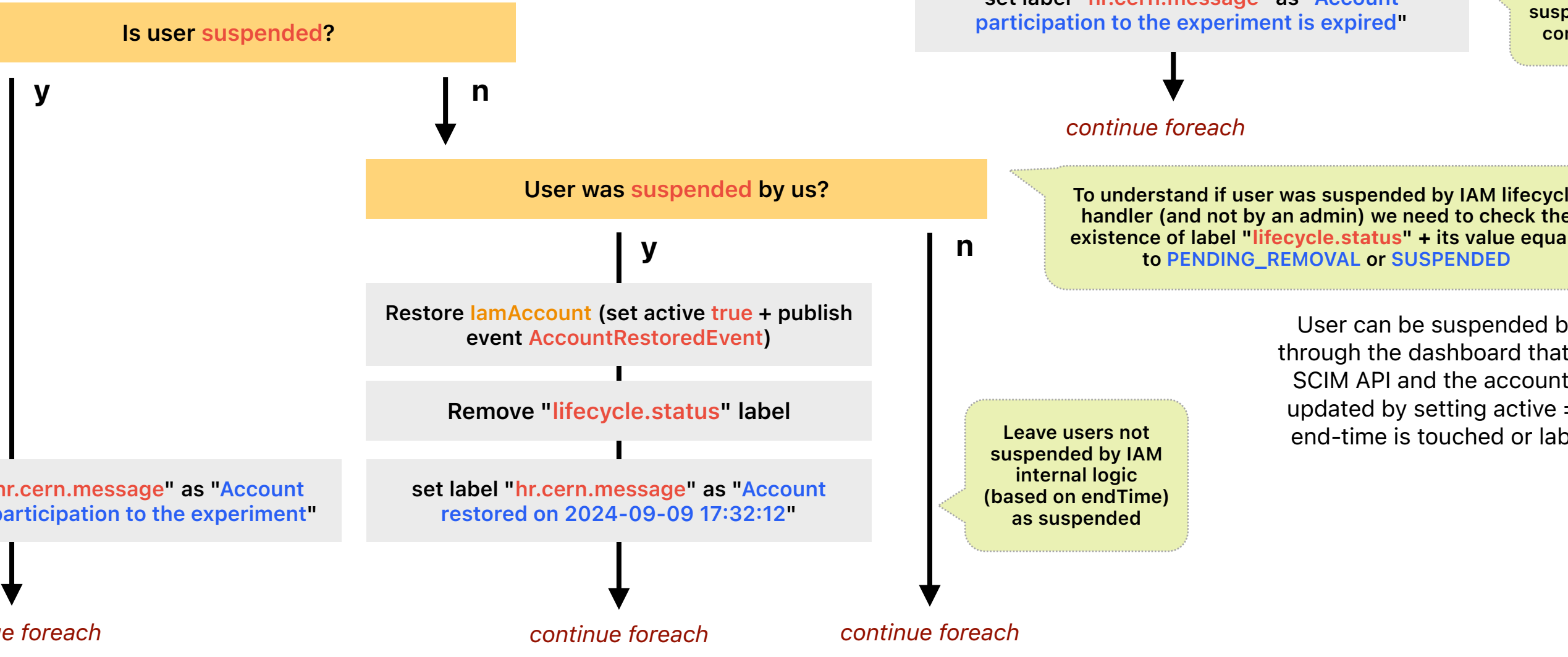>
> label "**hr.cern.action**" has been removed

We need a restore logic for non-CERN instances: setting the end-time could restore a user?

ExpiredAccountsHandler

retrieve the list of users which have an expired endTime

foreach expired **IamAccount**

is removal grace period expired + is remove expired accounts ?
- **y** → Add account to "**accountsScheduledForRemoval**"

  continue foreach
- **n** → is suspension grace period expired ?
  - **y** → disable account: set active **false**

    publish event **AccountDisabledEvent**

    is remove expired accounts ?
    - **y** → set label "**lifecycle.status**" as PENDING_REMOVAL

      set label "**lifecycle.timestamp**" as CURRENT_TIMESTAMP

      continue foreach
    - **n** → set label "**lifecycle.status**" as SUSPENDED

      set label "**lifecycle.timestamp**" as CURRENT_TIMESTAMP

      continue foreach
  - **n** → set label "**lifecycle.status**" as PENDING_SUSPENSION

    continue foreach

end foreach

Remove accounts in "**accountsScheduledForRemoval**"