

EGI Resource Allocation Procedure

Draft 1.0

Table of contents

EGI Resource Allocation Procedure	1
Definitions.....	1
Actors.....	1
OLA	1
SLA	2
User Request.....	2
Procedure description.....	3
Activities description.....	5
Activity 1 (for B): OLAs creation based on pool matching.....	5
Activity 2 (for P): OLA confirmation/rejection	5
Activity 3 (for P and B): OLA (re)negotiation step	5
Activity 4 (for B): SLA creation/updating	6
Activity 5 (for U and B): SLA negotiation step	6
Example of realistic scenario.....	7

Definitions

Actors

B – Broker (EGI)

P – either site or NGI, depending on Responsibility for the pool (NGI or Site) defined

U – User or VO manager

OLA

record in the RA database describing a specific site allocation associated with specific user request; parties that negotiate an OLA are: Site representatives (Site or NGI manager) and Broker (EGI).

OLA states:

- **DRAFT** – in this state OLA is visible only to its author
- **IN-NEGOTIATION** – state indicates that a proposal was sent by one party to another and a negotiation step is expected.
- **AGREED-REVOKABLE** – state indicates that OLA was created based on right-to-revoke scenario and it is in a period when it can be revoked.
- **AGREED** – state indicates that both parties agreed on the OLA, however it is not linked to any binding SLA; OLA in this state can be invalidated only in case the association with underpinned SLA is deleted both based on Broker or User decision).
- **BINDING** – this state indicates that resource allocation must be performed based on details described in the OLA
- **CANCELLED** – this state indicates that OLA was rejected and is neither binding nor a

subject of negotiation.

SLA

record in the RA database describing federated resources allocation for the VO (VO group) represented by the User; parties that negotiate an SLA are: Broker (EGI) and User; SLA must contain at least one *SLA section* (each section reflects a single underpinned OLA); SLA may contain metrics defined on federated level (there are direct obligations of EGI towards the User); SLA is created as a response to the RA User Request;

SLA states:

- **IN-NEGOTIATION** – state indicates that a proposal was sent by one party to another and a negotiation step is expected.
- **BINDING-IN-PART** – at least one SLA section is binding (also related underpinned OLA must be binding); this state is used in case some of the Site representatives agreed on their OLAs, but negotiations with others are still in progress.
- **BINDING** – this is terminal state for the SLA; this state indicates that RA process was completed (even in case that the request is not satisfied fully)
- **CANCELLED** – this state indicates that OLA was rejected and is neither binding nor a subject of negotiation

User Request

Initial request specified by user, specifying their resource requirements. A reply to this document is an SLA with underpinning OLAs.

Procedure description

EGI RA Procedure is to define steps/activities that will successfully lead from RA Requests specified by Users to binding federated SLAs and underpinning OLAs. All those activities will be supported by a tool which is subject on implementation in miniproject.

Due to many actors involved, this procedure cannot be defined as a simple sequence of activities. Our approach was to defined activities that can be performed by specific actors (User, Broker and Provider) under specific conditions. Conditions are related only to OLAs and SLAs states defined above. Activities are independent, atomic, and can be performed at any time by approved actors.

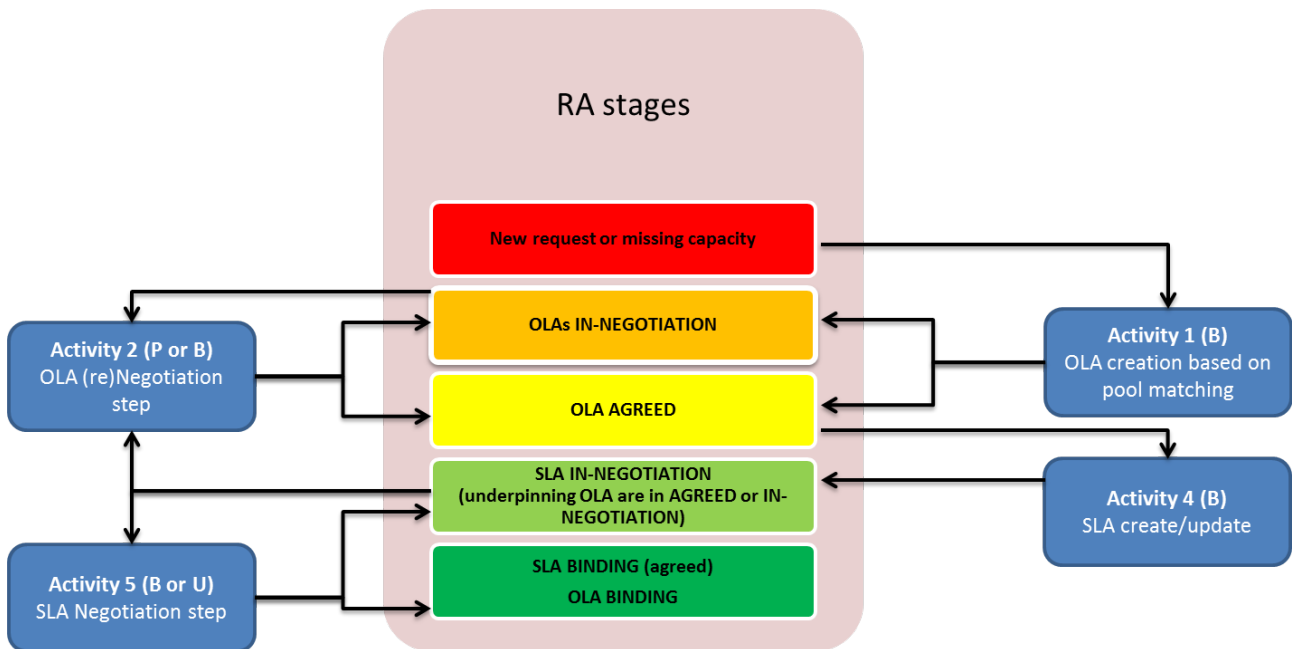


Figure 1: Stages of OLAs and SLAs in RA process

Figure 1 is an attempt to combine in a single picture such elements as: view on allocated resources on different stages (colorful rectangle in the center with its height representing capacity of requested resources and different colors representing stages of allocation), activities (blue boxes), and how activities change statuses of OLAs and SLAs (arrows). Due to better clarity, the Figure does not present CANCELLED state for both OLA and SLA. This state may be output of almost every activity. Figure 2 is an extension of Figure 1 with new state (AGREED-REVOKABLE) and an activity (Activity 3) to support right to revoke scenario.

Those pretty complicated pictures might become more clear after analysis of the example shown further in the document.

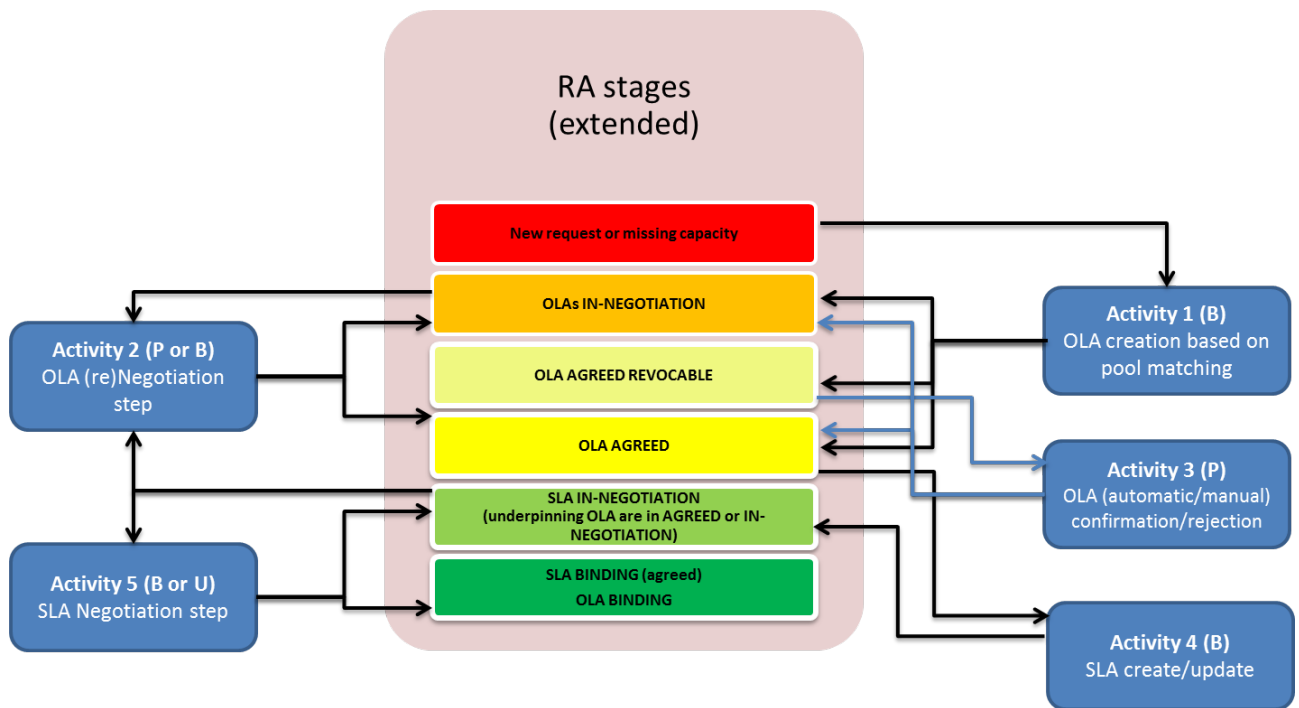


Illustration 2: Stages of OLAs and SLAs in RA process (extended version with revocable OLA)

Activities description

Activity 1 (for B): OLAs creation based on pool matching

This procedure might be run by Broker anytime when there is a need to add more resources to the pool.

Input/conditions:

- missing resources in a Request
- existing pools

Actions:

1. Broker runs a tool matching the pools with the Request
2. Selected offers from pools (created OLAs) are processed based on *scenarios* defined within the pools, according to following rules:
 1. *free hand* – OLAs are set as AGREED and notifications are sent to P
 2. *right to revoke* – OLAs are set as AGREED REVOKABLE and notifications are sent to P
 3. *negotiation* --- OLAs are set as IN-NEGOTIATION and proposal is sent to P

Activity 2 (for P): OLA confirmation/rejection

This procedure is valid only for the right-to-revoke scenario.

In case of no action, OLA is made AGREED automatically after revoke period.

Input/conditions:

- OLA is in the state AGREED REVOKABLE

Action:

- provider can confirm OLA, which makes OLA AGREED
- provider can reject OLA, which makes OLA REJECTED
- provider can propose changes to the OLA, which makes OLA IN-NEGOTIATION and proposal is sent to B

Activity 3 (for P and B): OLA (re)negotiation step

Note: PARTY is the actor who operates the procedure; OTHER PARTY means: P in case B runs the actions, or B in case P runs the actions;

Input/condition:

- OLA are IN-NEGOTIATION and proposal was received from OTHER PARTY

Actions:

- PARTY can agree on the proposal; then the OLA is set AGREED
- PARTY can reject the proposal; then the OLA is set to REJECTED
- PARTY can send a new proposal; then the OLA remains in IN-NEGOTIATION, the new proposal is sent to OTHER PARTY

Activity 4 (for B): SLA creation/updating

Input/condition:

1. at least one agreed OLA related to the request that is not covered so far by an existing SLA

Actions:

1. selected (all) agreed OLAs are linked to the SLA (if SLA did not exist, it is created)
2. Broker marks if this is a full offer or it can be extended in the future (it influencea the SLA state after negotiation success – BINDING-IN-PART or BINDING)
3. created/ updated SLA proposal is sent to the User

Activity 5 (for U and B): SLA negotiation step

Note: OTHER PARTY means: P in case B runs the actions, or B in case P runs the actions; PARTY is the actor who operates the procedure

Input/condition:

- SLA is IN-NEGOTIATION or BINDING-IN-PART and SLA proposal was received from OTHER PARTY

Actions:

- PARTY can agree on the proposal; then the SLA is set:
 - BINDING in case the proposal was marked as full;
 - BINDING-IN-PART in case the proposal was marked as extensible;
- PARTY can reject the proposal; then the underpinning OLA are set to CANCELLED state
- PARTY can modify proposal and send a new proposal; then the SLA remains in IN-NEGOTIATION state, the new SLA proposal is sent to OTHER-PARTY

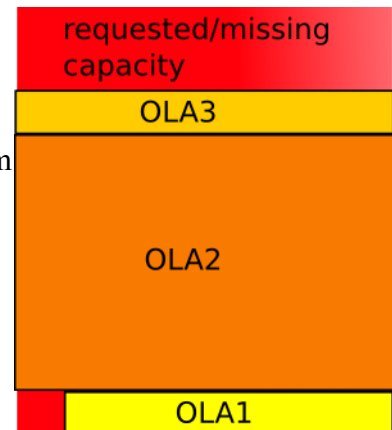
Example of realistic scenario

Situation of allocated resources after each step is presented at diagrams. Colour code is the same as in figure above in the document.

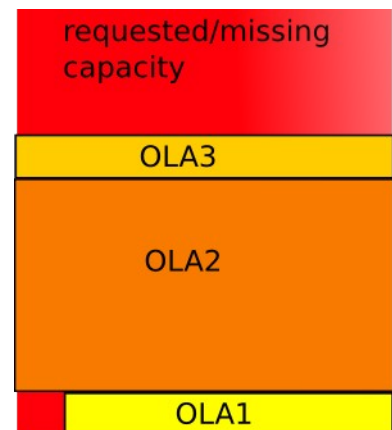
1. User specifies a request: 1000 cores from 1.08.2013 to 31.07.2014



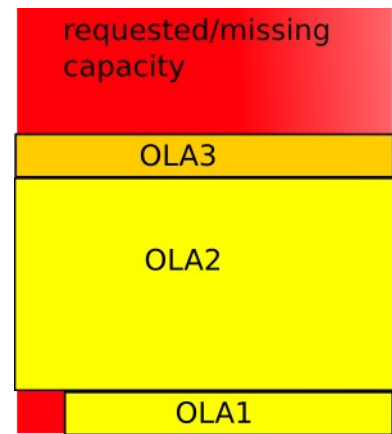
2. Broker executes Activity 1, which results with:
 - OLA1 with Site1 100 cores free hand from 1.10.2013 to 31.07.2014 (OLA1 is AGREED)
 - OLA2 with Site2 600 cores based on full negotiations from 1.08.2013 to 31.07.2014 (OLA2 is IN-NEGOTIATION, proposal was send to Site2)
 - OLA3 with Site3 200 cores right-to-revoke from 1.08.2013 to 31.07.2014 (OLA3 is AGREED-REVOKABLE for 7 days)



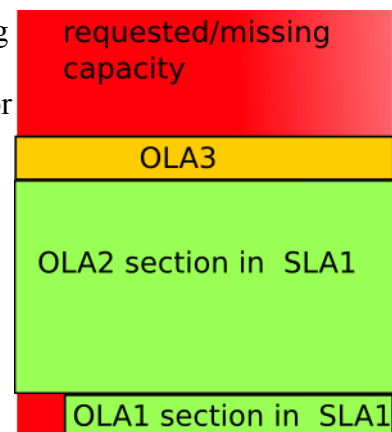
3. Site2 executes Activity 2 on OLA2, which results with OLA2 changed to 500 cores (OLA2 is IN-NEGOTIATION, proposal was sent to Broker)



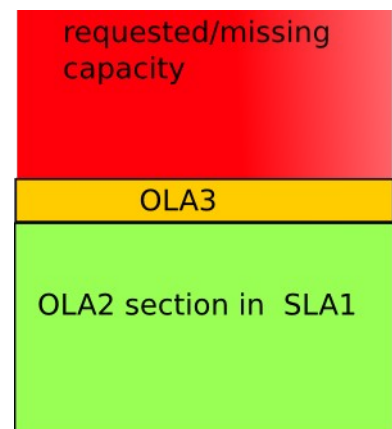
4. Broker executes Activity 2 on OLA2, which results in accepted OLA2 (OLA2 is AGREED)



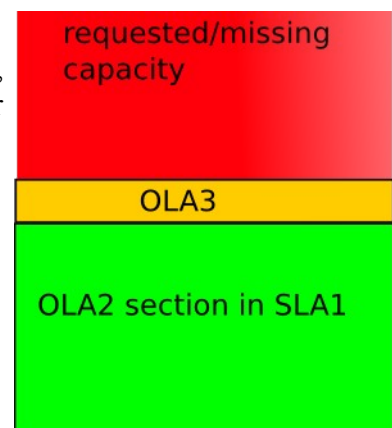
5. Broker executes Activity 4, which results in creating and sending to User SLA1 containing two SLA sections – related to OLA1 and OLA2, SLA1 is marked to be partial, as Broker still hopes for more resources based on OLA3 (SLA1 is IN-NEGOTIATION).



6. User executes Activity 5, which results with SLA1 with deleted SLA section related to OLA1 (it was essential for user to start 1.08). SLA1 proposal is sent by User to Broker. SLA1 is IN-NEGOTIATION



7. Broker executes Activity 5 accepting modification in SLA1; as the result SLA1 is BINDING-IN-PART, OLA1 is CANCELLED, OLA2 is BINDING, user can start computations (in case its later than 1.08.2013)



8. (after 7 days from creating OLA3, when OLA3 turned AGREED)
Broker executes Activity 4, which results in updating SLA1 with
SLA section based on OLA3; this time Broker marks SLA1 to be
completed (despite the fact that only 700 out of 1000 cores
requested where collected)

requested/missing
capacity

OLA3 section in SA1

OLA2 section in SLA1

9. User executes Activity 5 accepting the extention of SLA1, which
finalizes RA process (SLA1 is set BINDING, OLA3 is made
BINDING)

requested/missing
capacity

OLA3 section in SA1

OLA2 section in SLA1