



Figure 1: Diagram

- entity may have memberOf attributes if homeFederation matches Federation holding entity's data
- memberOf may have multiple values
- Federations can only ask directly trusted Federations for data

OP1 requests Edugate Federation (EF) for RP3's metadata

1. OP1 sends request to EF API
2. EF searches for RP3  $\Rightarrow$  not found
3. EF searches for memberOf tag of OP1  $\Rightarrow$  found Edugain
4. EF checks if Edugain is trusted by EF  $\Rightarrow$  true
5. EF sends request for RP3's metadata to Edugain; request must contain "Edugate Federation" to the list of visited Federations so Edugain can exclude it (avoiding loop)
6. Edugain searches for RP3  $\Rightarrow$  found
7. Edugain extracts RP3's data and searches for one of the following:
  - homeFederation and metadata/metadataLink
  - homeFederation

Found homeFederation  $\Rightarrow$  InCommon

8. Edugain sends request to InCommon for RP3's metadata including "Edugain" as some attr in the request

9. InCommon searches for RP3  $\Rightarrow$  found
10. InCommon validates value "Edugain" matches one of the values of memberOf  $\Rightarrow$  true
11. InCommon requests EntitiesRepository for RP3's metadata
12. EntityRepository finds RP3's data signs it and sends in response
13. InCommon validates signature, signs data with own sig and sends response to Edugain
14. Edugain validates signature, signs data with own sig and sends back to Edugate
15. Edugate validates signature, signs with own sig and sends data to OP

About RP4 InCommon would respond with error to Edugain as RP4 doesn't contain memberOf with Edugain value